DLT Labs

StarbaseToken Code Review

September 05, 2017

Table of Contents

1 Introduction

DLT Labs conducted a review of the smart contracts that make up the StarBase Token Sale service. The findings of the review are presented in this document.

This review was performed under a contracted fixed rate and performed by independent reviewers.

Update: The STARBASE Token Development team has informed us that they have deployed the fixes to the issues observed in our reviews and details of which are stated below. DLT Labs' independent reviewers have verified the fixes for the issues stated below and they can be marked as fixed from a review perspective.

1.1 Review Goals and Focus

Sound Architecture

This review includes both objective findings from the contract code as well as subjective assessments of the overall architecture and design choices. Given the subjective nature of certain findings it will be up to the StarBase Token development team to determine the appropriate response to each issue.

Smart Contract Best Practices

This review will evaluate whether the codebase follows the current established best practices for smart contract development.

Code Correctness

This review will evaluate whether the code does what it is intended to do.

Code Quality

This review will evaluate whether the code has been written in a way that ensures readability and maintainability.

Security

This review will look for exploitable security vulnerabilities.

1.2 Terminology

This review uses the following terminology.

Code Coverage Terms

Measurement of the testing code coverage.

1.2.1.1 Untested

No tests.

1.2.1.2 Low

The tests do not cover some set of non-trivial functionality.

1.2.1.3 Good

The tests cover all major functionality.

1.2.1.4 Excellent

The tests cover all code paths.

Severity Terms

Measurement of magnitude of an issue.

1.2.2.1 Minor

Minor issues are generally subjective in nature, or potentially deal with topics like "best practices" or "readability". Minor issues in general will not indicate an actual problem or bug in code.

The maintainers should use their own judgement as to whether addressing these issues improves the codebase.

1.2.2.2 Medium

Medium issues are generally objective in nature. Most medium level issues will not represent an actively exploitable bugs or security problem, but rather an issue that is likely to lead to a future error or security issue.

In most cases a medium issue should be addressed unless there is a clear reason not to.

1.2.2.3 Major

Major issues will be things like bugs or security vulnerabilities. These issues may not be directly exploitable such as requiring a specific condition to arise in order to be exploited.

Left unaddressed these issues are highly likely to cause problems with the operation of the contract or lead to a situation which allows the system to be exploited in some way.

1.2.2.4 Critical

Critical issues are directly exploitable bugs or security vulnerabilities.

Left unaddressed these issues are highly likely or guaranteed to cause major problems or potentially a full failure in the operations of the contract.

2 Overview

2.1 Source Code

The source code under review can be found in the public github repository https://github.com/StarbaseCo/STAR

2.2 Contracts

The reviews were primarily limited to below smart contracts.

- AbstractStarbaseCrowdsale.sol
- AbstractStarbaseMarketingCampaign.sol
- AbstractStarbaseToken.sol
- MultiSigWallet.sol
- StarbaseCrowdsale.sol
- StarbaseMarketingCampaign.sol
- StarbasePresaleWallet.sol
- StarbaseToken.sol

Zeppelin Base Contracts

The contracts make use of a small number of the re-usable contracts provided by Zeppelin. These contracts were not covered by this review.

3 Findings

This section contains detailed issues and analysis.

3.1 General Thoughts

The first review identified multiple issues that require patching prior to the contracts being deployed and used. The first review led to identification of no Critical Issue, but three Major and two Minor issues were reported.

A second review was conducted to further review the previously identified issues and also see if any new issues were created due to code update. No more new issues were observed during the second review.

Code Quality

The code is generally of good quality. Commonly used logic for token sale has been encapsulated in internal utility functions. The Token contract is ERC-20 compatible and uses standard Zeppelin libraries. The Zero address attack and Short address have been handled appropriately.

3.2 Critical Issues

No critical Issue was observed during the review of the smart contract codes.

3.3 Major Issues

Issue 1: In smart contract StarbaseCrowdsale.sol, logic written in line number – 590 to 595 of function recordPurchase(), lets user get STAR token for free in particular edge case scenario. The edge case scenario is explained below.

Explanation: Suppose current totalAmountOfCrowdsalePurchases is 59M CNY and User purchases STAR token corresponding to 5M CNY at bonus tier 20%. Then total amount invested including bonus would be (5M+1M) = 6M CNY. Now totalAmountOfCrowdsalePurchases would become 65M CNY i.e. more than MAX_CROWDSALE_CAP of 60M CNY. Now we will initiate refund of ether before recording investment. So as per logic written, difference will be calculated as (totalAmountOfCrowdsalePurchases - MAX_CROWDSALE_CAP) that is 5M CNY and user will be refunded Ether of 5M CNY in his account and user's purchase will be recorded for remaining 1M CNY.

In this way, User can purchase token of 1M CNY with full refund of investment of 5M CNY.

Action: Fixed

Issue 2: In StarbaseMarketingCampaign.sol at line no. 72, function withdrawRewardedTokens() can be called by anybody. It means anybody can initiate withdraw reward token for any contributor. It should be checked via modifier or contributor address should be taken from transaction initiator based upon business logic.

Action: Fixed

Issue 3: In StarbaseMarketingCampaign.sol at line no. 120 at to assert condition of function updateRewardForContributor(), reward cannot be updated for contributor if all existing rewards are already withdrawn.

Action: Fixed

3.4 Minor Issues

Issue 1: In StarbaseCrowdsale.sol at line number – 274 of function endCrowdsale(),because of assert condition the timestamp of recordPurchase can hold time later than endedAt timestamp. Detailed explanation below

Explanation: While calling function endCrowdsale(), owner must have pass timestamp less than current timestamp. endedAt will be updated with this passed timestamp while purchase will be still going on. So purchase can be recorded of later timestamp than endedAt. Although it is not having any negative impact on crowdsale but it is advisable to rectify this.

Action: Starbase team decided not to proceed with the fix on issue as it has no negative impact on the crowdsale.

Issue 2: Inconsistent statements was found in statements related to limit of extended sale in page no.48 and 49. Although whitepaper talks about limit extended sale and same will be immediately finished when goal (2.5M or 5M whichever is correct) is reached. No logic was written for above in Smart Contract.

Action: Starbase decided not to add such mechanism as the raised funds already reached minimum investment cap from other partner Bitcoin suisse and early purchasers.