BnkToTheFuture.com Smart Contracts Audit by ZK Labs

MATTHEW DI FERRANTE, NICK JOHNSON

2018-01-30

Introduction

On 2018-01-30, Nick Johnson & Matthew Di Ferrante performed an audit of the BnkToTheFuture.com smart contracts. My findings are detailed below.

ZK Labs have no stake or vested interest in BnkToTheFuture BFT SP. This audit was performed under a contracted rate with no other compensation.

Authenticity

This document should have an attached cryptographic signature to ensure it has not been tampered with. The signature can be verified using the public key at http://keybase.io/mattdf

Audit Goals and Focus

Smart Contract Best Practices

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

Code Correctness

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

Code Quality

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

Security

This audit will look for any exploitable security vulnerabilities, or other potential threats to either the operators of ChainLink or its users.

Testing and testability

This audit will examine how easily tested the code is, and review how thoroughly tested the code is.

About BnkToTheFuture.com

BnkToTheFuture.com is an online investment platform for investing in FinTech, Bitcoin and Blockchain companies that plans to utilise the BF Tokens (BFT). Bnk To The Future BFT SP is the issuer and developer of the BFT Smart Contract.

Terminology

This audit uses the following terminology.

Likelihood

How likely a bug is to be encountered or exploited in the wild, as specified by the OWASP risk rating methodology.

Impact

The impact a bug would have if exploited, as specified by the OWASP risk rating methodology.

Severity

How serious the issue is, derived from Likelihood and Impact as specified by the OWASP risk rating methodology.

Overview

Source Code

The BnkToTheFuture smart contract source code was made available in the bnktothefuture/bft Github repository.

The code was audited as of commit `c17cff0e4a394b2d676d3246cec928df3ca75396`.

The following files were audited:

```
826a9f58e88602d9618f594931a259f7c73b7f46401f5a8120feb11498cb5837  BftCrowdsale.sol
41946d28418ea8b5a0b6bbe5c9f44abf923949052f52c27877f6f3dacb35be4e  BftToken.sol
```

The code makes extensive use of OpenZeppelin library code, which was *not* audited as part of this audit.

General Notes

The code is generally well structured and easy to read. It makes extensive use of the OpenZeppelin smart contracts, which reduces the count of lines that need to be independently audited and the risk of bugs.

Contracts

`BftToken` implements an ERC20 token, with additional functionality for upgrading to a new token in the future. It implements `CappedToken` and `PausedToken`, and thus has an owner, who can halt the token contract if ordered by the owner. The token ownership can only be transferred to an external address once the crowdsale is over.

`BftCrowdsale` implements a crowdsale mechanism for selling tokens during a fixed period. It includes functionality to preallocate a subset of tokens to certain parties with a fixed lock duration. The token price is adjustable prior to the crowdsale `start` date, and fixed beyond then.

Testing

Tests are run via a single shell script. We were able to independently confirm the validity of the tests.

Test coverage is fairly good, covering the happy path and likely errors.

Automated builds have been enabled for the repository, reducing the risk of accidentally adding errors.

Findings

We found 2 note issues, and no higher severity issues.

Note Issues

Pausable token allows token owner to halt trading

- Likelihood: low
- Impact: low

`BftToken` implements `PausableToken`, which allows the owner of the token contract to halt all trading of tokens.

As a result, the owner must be trusted to some degree not to interfere with operation of the token contract unnecessarily.

Sale participants cannot buy tokens if they already hold tokens

- Likelihood: low
- Impact: low

`BftCrowdsale` does not allow `buyTokens` to succeed if the beneficiary address already has more than 0 tokens, even if the number of tokens they have is below the maximum per-buyer cap. This is unorthodox and does not seem to serve any real purpose.

Low Issues

None found.

Medium Issues

None found.

High Issues

None found.

Critical Issues

None found.