# Security Audit of the “Best Millions” ICO contracts

## Contents

## Overview

Following document represents security audit of the contract suite provided by Best Millions ICO. Following document will provide review of the code and given specification and give, if necessary, advices on how to improve given system.

## Purpose of report

The scope of our review is limited to a review of Solidity code and only the source code we note as being within the scope of our review within this report. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The Solidity language itself remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond Solidity that could present security risks.

The report is not an endorsement or indictment of any particular project or team, and the report does not guarantee the security of any particular project. This report does not consider, and should not be interpreted as considering or having any bearing on, the potential economics of a token, token sale or any other product, service or other asset.

No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. Specifically, for the avoidance of doubt, this report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project.

## Methodology

The review was conducted during 2018-Jan-08 thru 2018-Jan-12.

Procedure can be summarized as follows:

1. Code review

   Manual review of code

   Comparison to specification

2. Testing

   Test analysis

   Functional testing

3. Best-practices review
4. Itemize recommendations

## Specification

Understanding of the specification was based on the description of the token sale provided in the README.md file in the github repository at the time of the audit. Also this audit takes into consideration white-paper given on [www.bestmillions.com](www.bestmillions.com) ICO web page on the date 2018-Jan-10.

Audit recognizes following specifications:

1. ICO is crowdsale that is time limited and capped. Crowdsale is refundable with defined soft cap under which all invested ethereum will be refunded to the investors.
2. ICO Token is defined under name Best Millions token (BMT), has 18 decimals and after crowdsale ends, it won't be possible of minting more tokens.
3. ICO token is burnable.
4. ICO token will base its value on buyback price defined by profit of the ICO after defined period of time.
5. BuyBack functionality is defined as guaranteed price offered by Best Millions ICO for all investors who want to sell back their tokens. After trade is finished, token sold are burned.

## Source code

Audit covers source code given on gitHub repository github.com/BestMillions with the latest commit `2c8344ce3755e4a02776c7751d81ac13bf65f085`. Audit covers two smart contracts named BestMillionsCrowdsale.sol and BestMillionsToken.sol representing crowdsale and token contracts in that order.

## Code review

Objective of this audit is to evaluate the Best Millions ICO code for security-related issues, code quality and adherence to best-practices.

Possible issues include (but are not limited to):

Transaction-ordering dependence

Timestamp dependence

Mishandled exceptions and call stack limits

Unsafe external calls

Integer overflow / underflow

Reentrancy and cross-function vulnerabilities

Denial of service / logical oversights

**BestMillionsToken** contract represents ERC-20 compatible token contracts. It defines name, symbol and number of decimals that are defined in specification. BestMillionsToken contract inherits behavior defined by CappedToken, PausableToken and BurnableToken contracts in that order. These contracts are part of the Open-Zepelin framework (www.openzeppelin.org/) and as such are audited and subject of this audit. By using these contracts, BestMillionsToken is fulfilling specifications of the ICO stating burnable and mintable token. Token contract doesn't override any default behavior of inherited contracts. Token contract is defining additional functionality required by buyback specification. This functionality is defined by sellBack and setRate functions.

**Recommendation**: sellback function (defined as: sellBack(uint256 _amount) public whenNotPaused) is requiring that balance of the contract must be greater than or equal to amount of ether that is payed for given tokens. There is no feedback information to the owner of the tokens or the contracts if condition is not met. Audit recommends creating event that will be fired every time this situation occurs.

As this token contract serves as BuyBack fund it has deposit and withdraw function. Withdraw function is properly secured by onlyOwner modifier.

Token is defined as pausable which is considered best practice of having emergency pause switch to prevent any damage or potential loss of funds.

**BestMillionsCrowdsale** contract functions as fund raising contract. It defines cap, goal and time period. It inherits behavior from CappedCrowdsale and RefundableCrowdsale contracts also part of the Open-Zeppelin framework. They provide behavior defined by specification. Finalize action is redefined and has following flow: mint residual tokens, stop minting and unpause token. Mint of residual token uses TimeLocked token contracts with predefined addresses that control release time of specified amount of tokens given to those addresses. TokenTimelock is also part of the Open-Zeppelin framework and not subject of this audit.

Calculate rate internal function uses SafeMath which is considered best practice.

## Code review conclusion

Audit didn't find any security-related issue. Both contracts are written with best-practices implemented and by using tested and audited libraries.

## Testing

### Test analysis

Test defined in ICO repository provide extensive coverage of all possible paths. Given tests cover not only two ICO contracts but also Open-Zeppelin contracts used in ICO contracts. Given tests are well written and audit doesn't propose any changes.

### Functional testing

Process of this audit includes functional testing of the given contracts. Audit covers following cases and actions:

TC1: Actions before start time of the ICO:

State: ICO contracts deployed, crowdsale not yet started

Actions/Expected outcome/**Tested outcome**: buying tokens/forbidden/**forbidden**, minting or burning tokens/forbidden/**forbidden**, finalizing crowdsale/forbidden/**forbidden**.

TC2: Actions during crowdsale:

State: ICO contracts deployed, crowdsale in progress.

Actions/Expected outcome/**Tested outcome**: buying tokens/allowed/**allowed**, trading-transferring tokens/forbidden/ **forbidden**, selling back tokens/ forbidden/ **forbidden**.

TC3: Actions after sucesfull crowdsale:

State: ICO contracts deployed, crowdsale finished with goal reached.

Actions/Expected outcome/**Tested outcome**: buying tokens/forbidden/ **forbidden**, trading-transferring tokens/allowed if crowdsale finalized/**after conditions have been met allowed**, finalization of the crowdsale/allowed only to the owner/**allowed only to the owner**, selling back /allowed if crowdsale finalized/**after conditions have been met allowed**.

TC4: I Actions after sucesfull crowdsale:

State: ICO contracts deployed, crowdsale finished with goal not reached.

Actions/Expected outcome/**Tested outcome:** Refund of funds/allowed/**allowed**.

Given cases were tested and behavior and results were matching with excepted. Summary of functional tests is listing only relevant actions for the given test case.

## Recommendations

### Code documentation

Current ICO contracts are lacking documentation. Auditor recommends creating documentation for all functions following established rules sing documentation tags @param, @dev,@returns.

### Compiler warnings

The solidity compiler reported three warnings about hard-coded addresses in BestMillionsCrowdsale.sol. To clear these warnings, use checksummed addresses.

Reference: how to specify a hard-coded address as a literal

## Appendix

### File Signatures

Below are file signatures of the relevant files reviewed in the audit.

/contracts

```
3aa06136799ddb9eb3c0d800d280e45f BestMillionsCrowdsale.sol

111fcf4c467bcc9d0fc30487481cf2e2 BestMillionsToken.sol

a8ff10350a5fc0f30c28d92f872f2320 Migrations.sol
```

/test

```
543ba844dc400722cbc07b4e1c2029a0 BestMillionsCrowdsale.test.js

737cec336657c0288cb49c153f6d94f4 BestMillionsToken.test.js

b43fc2821a580898727038339977cf70 CappedCrowdsale.test.js

e0695aff0ffd190fa38339e3c558020f Crowdsale.test.js

0bc84ca32968856ae68c652e5882df2f FinalizableCrowdsale.test.js

ec2d15d2b4cf2b4f2e609d807fc8af3b RefundableCrowdsale.test.js

b0700341a51100284e45999600755c55 TokenTimelock.test.js
```

| Date of change | Message | Editor |
|---|---|---|
| 12.01.2018. | Document created | S.Pantelic |
|  |  |  |
|  |  |  |
|  |  |  |