CommerceBlock Solidity Contracts Audit

AUTHOR: MATTHEW DI FERRANTE

2017-10-25

Audited Material Summary

The audit consists of the following contracts:

- BasicToken.sol
- CommerceBlockToken.sol
- ERC20.sol
- ERC20Basic.sol
- MultiSigWallet.sol
- SafeMath.sol
- StandardToken.sol

The git commit hash of the reviewed files is `9fe5e0a4f6a29d742b1e49bcfddd498a7442abcc`.

Description

The contracts are all standard copies of the code found in the OpenZeppelin contracts, and the `MultiSigWallet` by Gnosis. They do not implement a crowdsale, merely a token with the entire initial balance owned by the company address.

There are no security flaws in the contracts. Other than the `CommerceBlockToken` contract which is an initializer contract for `StandardToken`, all other contracts are taken from the `zeppelin-solidity` github repository.

CommerceBlockToken.sol

This contract simply extends StandardToken and initializes name, symbol, decimals, and totalSupply variables. It also sets the initial balance to the company address in the constructor, and that address will distribute the funds after the contribution period.

The total supply allocated is 1 billion CBT Tokens.

BasicToken.sol

This is `BasicToken` from OpenZeppelin's token/BasicToken.sol with no functional changes.

ERC20.sol

This is just an ERC20 interface contract that meets the standard defined in:

https://github.com/ethereum/EIPs/issues/20

There are no security issues as it has no runnable code.

ERC20Basic.sol

This contract is again simply an interface to a simpler ERC20 Token interface that defines a `Transfer` event, `balanceOf` and `transfer` prototypes, and a `totalSupply` getter.

This is functionally equivalent to `ERC20Basic` from OpenZeppelin's token/ERC20Basic.sol

MultiSigWallet.sol

This is an unmodified copy of Gnosis' MultiSigWallet, SHA256 hash:

`77c9e4f54039392ed96f66ccb227e827d919c32ea760a671f7f0c6b3f8f7aa27`

as found in: https://github.com/gnosis/MultiSigWallet/blob/master/contracts/MultiSigWallet.sol

SafeMath.sol

This is OpenZeppelin's `SafeMath` contract with the `mul` and `div` functions removed, as they are not used:

```
$ diff zeppelin-solidity/math/SafeMath.sol cbt-smart-contracts/contracts/SafeMath.sol

8,13d7
< library SafeMath {
<   function mul(uint256 a, uint256 b) internal constant returns (uint256) {
<     uint256 c = a * b;
<     assert(a == 0 || c / a == b);
<     return c;
<   }
15,20c9
<   function div(uint256 a, uint256 b) internal constant returns (uint256) {
<     // assert(b > 0); // Solidity automatically throws when dividing by 0
<     uint256 c = a / b;
```

```
<     // assert(a == b * c + a % b); // There is no case in which this
   doesn't hold
<     return c;
<   }
---
> library SafeMath {
31a21
>
```

StandardToken.sol

This contract is, again, just OpenZeppelin's `StandardToken` contract with no functional changes, only comment and convenience variable changes compared to the current master branch of OpenZeppelin.

Disclaimer

This audit concerns only the correctness of the Smart Contracts listed, and is not to be taken as an endorsement of the platform, team, or company.

Audit Attestation

This audit has been signed by the key provided on https://keybase.io/mattdf - and the signature is available on https://github.com/mattdf/audits/