IMMLA Presale Token Smart Contract Audit by Ambisafe Inc.
July, 2017
Tanya Bushenyova, Oleksii Matiiasevych

1. **Introduction.** IMMLA requested that Ambisafe perform an audit of the contract implementing their token presale logic. The contract in question is hosted at:

   https://github.com/IMMLA/PresaleToken/blob/master/contracts/Immla.sol

   Contract in scope:

   - PresaleToken

2. **Disclaimer.** The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bugfree status. The audit documentation is for discussion purposes only.

   We have not audited IMMLA's key management operational security; this could impact the safety of the IMMLA presale smart contract. Compromising of token manager account could result in all collected ETH being locked inside the contract.

3. **Executive Summary.** Contract code is kept simple and as such increases confidence. There is only one external call from the contract which is sending ether to escrow account. Escrow account must be tested for the possibility of accepting ETH through a send() function, otherwise, if it will fail to receive ETH, all the collected funds will be locked.

4. **Critical Bugs and Vulnerabilities.** No places in code were identified as critical issues.

5. **Line By Line Review.**

   5.1. Line 84. Since there is no iterable array of owners (only balances mapping) and the crowdsale manager will iterate over all owners externally using burnTokens(_owner) function then there's a possibility that some owner will be skipped and the contract will be stuck in Migrating stage for a prolonged period of time. Keeping track of all owners is now performed only outside of the contract using the LogBuy events. Consider keeping an array of owners inside the contract so that it would be easier to track which balances were not migrated.
   Note: if the number of presale participants will be small, there won't be any complications with current implementation.

5.2. Line 144. Minor: needs to be used with extreme care, as calling it with setTokenManager(%wrongAddress%) will result in a permanent loss of control over the contract. Consider 2 steps ownership transfer or at least add sanity check for 0 address, as in constructor.