V1.0

AidCoin

Audit

Dean Eigenmann - Zero Knowledge Labs
January 14, 2018

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.

Overview

The audit was performed on all smart contracts provided by the AidCoin team, the contracts are found in the directory: contracts.

Audit Results

No major issues were detected in the code, however there is a function that should be removed. Additionally some minor optimizations can be made.

AidCoin.sol

This contract is the main token contract, it inherits from the Burnable & Mintable token contracts.

Modifiers

canTransfer

This modifier ensures that minting has been finished.

Suggestions

1. Remove arguments as they are never used.

transfer

This function adds the canTransfer modifier, and then calls the transfer function of the parent contract.

transferFrom

This function adds the canTransfer modifier, and then calls the transferFrom function of the parent contract.

AidCoinIco.sol

This contract is responsible for the main crowdsale logic.

The contract inherits from the Crowdsale and Ownable contract and uses SafeMath for uints.

Constructor

This function calls the parent construct. The token is set from an address that was passed.

buyTokens

This function is responsible for the main sale logic.

The function ensures that the passed beneficiary is not the burn address and that the purchase is valid.
The transferred wei amount is multiplied by the rate to calculate the token amount.
It is ensured that the amount of tokens is a valid amount.
The amount of tokens is added to the sold token amount.
It is ensured that the sold token amount does not exceed the token cap.
The amount contributed is added to both the weiRased.
The soldTokens variable is updated.
The tokens are minted for the beneficiary address.

updateEndDate

This function allows the owner of the contract to extend the crowdsale time.

Suggestions

1. Remove this function as it allows the owner to be malicious and continuously extend the endTime of the crowdsale.

closeTokenSale

This function transfers the owner of the token once the sale has ended.

Suggestions

1. Call the finishMiniting function in this contract, this will ensure the owner cannot be malicious and mint more tokens than promised.

changeParticipationLimits

This function allows the owner to change the maxWithoutWhitelist and the maxWithWhitelist values of the crowdsale contracts.

changeWhitelistStatus

This function allows the owner to set the values of the mapping whitelist for an array of users to a particular amount.

Suggestions

1. It should be ensured that the length of the array of users and amounts are identical.

checkValidAmount

This function ensures that the bought amount of token does not exceed the single cap per user. Also it is ensured that it does not exceed the whitelist or non whitelist limits. Once done, it updates the boughtTokens mapping for the specified beneficiary.

hasEnded

Ensures that the cap has been reached, or that the current time is greater than the end time.

hasStarted

Ensures that the current time is in between the start and end time.