SOLIDIFIED

Audit Report for Kuende October 17, 2018.

Summary

Audit Report prepared by Solidified for Kuende covering the token and crowdsale contracts.

Process and Delivery

Two (2) independent Solidified experts performed an unbiased and isolated audit of the below token sale. The debriefing took place on October 17, 2018, and the final results are presented here.

Audited Files

The following files were covered during the audit:

- KuendeToken.sol
- KuendeCrowdsale.sol

Notes

The audit was based on the solidity compiler `0.4.25+commit.59dbf8f1`
The audit was performed on commit `9d680349b3cfd8d835ce8d927f14f0ea67f62c38`

Issues Found

Minor

1. Investors can buy tokens below minimum investment

In cases where an investor sends ETH that exceeds either the *contract cap* or the *investor cap*, they can buy tokens with an amount less than `minInvestment` with their remaining ETH. This is because of the validation which checks whether the wei amount is greater than zero instead of the minimum investment.

Recommendation

It is recommended to validate the remaining amount after the refund against the minimum investment.

In `buyTokens()` line 195

```
require (weiAmount > 0 );
```

Should be changed to

```
require (weiAmount >= minInvestment);
```

2. The modifier *notEnded* does not consider the end time

As per the comment included in the code, the investment duration includes both start and end time specified in the contract. The modifier will incorrectly fail if the current time is exactly equal to `endTime`.

Recommendation

The value can be negligible, but it is recommended to include the end time while calculating.

```
require(now < endTime)
```

Should be changed to

```
require(now <= endTime)
```

Notes

3. Provide meaningful error messages for every exception

It is recommended to provide meaningful error messages along with each `require` statement. This will help the user to understand what went wrong more easily since there are many validations happening for each buy.

4. Use Withdrawal pattern

Use of withdrawal pattern can be considered to refund the excess amount to the buyer. This is not necessarily an issue since DoS caused by the buyer affects only their transaction, but it may cause a denial of service if contracts without a payable fallback from buying not-exact values, because the `transfer()` to the contract in `RefundExcess()` will revert. For more information refer this.

5. Avoid duplicate validations

Multiple duplicate validations are present in the code. For example, crowdsale start time is validated in multiple locations. Remove them to save gas.

6. Storing token balance can be avoided

The value `tokenBalance` is completely irrelevant to how the smart contract functions and does not need to be stored. Consider removing it if it was not intended.

7. No need to update weiAmount while refunding

Updating weiAmount on line 264 is completely irrelevant to `refundExcess()` functioning correctly. Consider removing it.

8. Consider adding a function to calculate the token amount

Adding a function to calculate the token amount can help the buyer to check the number of tokens that can be bought. Use this same function to calculate the token in the `buyTokens()` function.

9. The contract does not take advantage of storage gas refunds

The contract needs a function that gets called after the crowd sale is over in order to reclaim storage gas refunds. Gas refunds are discussed in section 6 of Ethereum's Yellow paper.

Closing Summary

Several issues were found during the audit which could break the intended behaviour. It is recommended for the Kuende team to address the issues. It is furthermore recommended to post the contracts on public bounty following the audit.

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of the Kuende platform or its products. This audit does not provide a security or correctness guarantee of the audited smart contracts. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Solidified platform from legal and financial liability.