KEYis Smart Contracts Audit Report

The Code is located in theKEYis github Repository and the version used for this commit is. e323342db359e18f532392b0d4d4d733ff4b0566

23rd December 2018

Security Level references

Every issue in this report was assigned a severity level from the following:

High severity issues will probably bring problems and should be fixed

Medium severity issues could potentially bring problems and should eventually be fixed.

Low severity issues are minor details and warnings that can remain unfixed but would be better fixed at some point in the future.

High severity issues:-

1) Token amount is not multiplied by 1 ether or 10^18 to get sufficient amount of tokens, that's why after calling buyToken function by whitelisted user having value 1 ether, investor is not be able to get 2000 tokens, they are only getting `0.00000000000002000`

Status : Not fixed yet

Recommendations :

```
investorAlloc = ((totalSupply * 75) / 100) / 1 ether;
teamsAlloc = ((totalSupply * 15) / 100) / 1 ether;
costsAlloc = ((totalSupply * 10) / 100) / 1 ether;
standardToken = 2000 ;
```

It should be like as shown below, as after dividing with 1 ether amount investor allocation will be get 1/10**18

```
(totalSupply * 75) / 100)
 standardToken = 2000 * 1 ether;
```

If you want for display purpose to show value without decimals you can implement a getter function that will display amount after dividing by 1 ether.

2) Test case fail of approving negative tokens, it should not approve negative tokens.

Status : Not fixed yet

Recommendations : use `require(_value > 0)`, and check if allowance value is zero or not before performing operation,

Also use function increase and decrease approval to change allowance .

Medium Severity Issues:-

1) Avoid state change after transfer function or external calls token sale contract line no 225 must be before transfer function.

Low Severity Issues:-

1.) Solidity version must be fixed(Always use latest Version).
It should not **pragma solidity ^0.4.23;**
It should be **pragma solidity 0.4.23;**
Status : Not yet Fixed.

2.) you can remove event emit at line no 229 as already transfer function contain event emit, at etherscan event is listen and it's showing ERC20 token is transferred to two address, it means it's listening two transfer event even though both are same still you can remove from line no 229.

3.) your specification document contain compiler version 0.4.21, and you are using ^0.4.23 make sure you are using same resources, compilation version and libraries that are listed in your paper.

Unit Testing

Test Suite Result

✓ Should correctly initialize constructor values of Contract (662ms)

✓ Should whitelist accounts[1] (123ms)

✓ Should buy correct Tokens (249ms)

✓ pause and get sale status (216ms)

✓ unpause and get sale status (149ms)

✓ Should Chnage owner (159ms)

✓ Should Burn tokens (128ms)

✓ Should not Burn Negative tokens (101ms)

✓ Should Enable mannual tier (139ms)

✓ Should be able to transfer Tokens from one beneficeary to another when token lock period is over (196ms)

✓ should Approve address to spend specific token (103ms)

❌ should not Approve address to spend negative tokens (79ms)

✓ should Transfer from account 1 to account 3 (256ms)

✓ Should remove whitelisted accounts[1] (146ms)

✓ Should be able to end sale (344ms)

Implementation Recommendations

Contract doesn't implement the correct way to send tokens to the investor, some of the test cases are left implementing because of 1st high severity issue, as even test faucet(ether) will not be able to buy all the tokens, so testing of tier switch is left and will only be possible when all issues listed above will be solved.

Also as a investor i would like to know if your dashboard for investors are ready or not or you want investor to use metamask directly ?, because in your smart contract, investor even after getting whitelisted will not be able to get tokens by directly sending ether to tokenSale contract, they need to call buy tokens directly that's why you need to share your source code with every investor to directly call function buyToken(), so to avoid this you can remove revert() from fallback function and call buyTokens().

Comments:-

Overall, the code is clearly written, and demonstrates effective use of abstraction, separation of concerns, and modularity. KEYis development team demonstrated high technical capabilities, both in the design of the architecture and in the implementation.
We found some critical issue and several additional issues that require the attention of the KEYis team. Given the subjective nature of some assessments, it will be up to the KEYis team to decide whether any changes should be made.

Transaction Hash of all the test performed on test network (Rinkeby).

https://docs.google.com/spreadsheets/d/1NJY_mLfpPCzsbpD6l1DfgaB4whxcXofxdk4dTfN_c5Q/edit?usp=sharing