PAY ACCEPT

https://github.com/PayAccept/token/blob/master/PAYtoken.sol

CONTRACT AUDIT

By

Sheraz Arshad

Introduction

In this Smart Contract audit we'll cover the following topics:

1. Disclaimer
2. Overview of the audit and nice features
3. Critical vulnerabilities found in the contract
4. Medium/Low severity vulnerabilities found
5. Line by line comments
6. Summary of the audit

1. Disclaimer

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

2. Overview

The Smart Contract comprises standard **ERC20** functionality in addition to sub-functionalities of **Crowdsale** and **Airdrop** of the **PAY** tokens. The contract is **Ownable** which means that it has several functions which can only be transacted by the owner of the contract. The complete contract has **653** lines of Solidity code.

The purpose of this Smart Contract is to implement **PAY** token based on the **ERC20** standard. The contract also serves the purpose of **Crowdsale** of **PAY** tokens, which is initiated by the owner of the contract by triggering the change in status of the contract to start of **ICO**.

The contract also implements **Airdrop** functionality which is transacted by the owner to do bulk transfer of tokens. The recipients receive balance in Airdrops which is deducted from the owner's balance itself.

The Smart Contract contains features such as black of listing addresses, for whom we want to restrict any meaningful interaction with the contract. We are also able to change the price of tokens per Ether that are sold in the **Crowdsale**. Both of these features can only be executed by the owner of the contract. In addition to these, the owner has the authority to start and end **ICO** stages of **Crowdsale**.

Nice Features:-

- Has a **blackList** function to blacklist any address from interacting with the contract.
- All the functions emit their respective events which helps us for better tracking of historical data as well as serves the purpose of better integration with blockchain explorers.
- Has **doAirdrop** function to airdrop tokens to list of addresses.
- Has **increaseApproval** and **decreaseApproval** functions to safeguard against running attacks of setting allowances.

- Has functions to change stages of ICO from **none** to **start** and **start** to **end**.
- Owner is able to change the price of tokens per Ether.
- Blacklisted addresses are not allowed to participate in **Crowdsale** or transfer of tokens. Essentially, their funds are locked once they are blacklisted and the funds cannot be moved until their status of blacklisted is changed.
- The Ethers that are received as part of **Crowdsale** are immediately transferred to the owner of the contract at the time of the user's participation and the users get their bought tokens based on the current **basePrice**.
- The functions of **transfer** and **transferFrom** return appropriate **bool** values at the end of their executions, which make their compatibility with wide ranges of exchanges and third-party Smart Contracts possible.
- Makes use of **SafeMath** when doing all the mathematical operations. This safeguards us against any underflow and overflow issues of integers.

3. Critical vulnerabilities found in the contract mint/airdrop/pickWinner/transfer/transferFrom

The contract has no critical vulnerabilities.

4. Medium/Low severity vulnerabilities found

Some of the functions in the codebase lack **natspec** comments which are very helpful for quickly understanding the functionality of the contract.

All of the functions have their visibility specified, so it is safe in that regard.

5. Line by line comments

There are not enough wide ranges of issues to warrant line by line comments. Every function can be reviewed if there are **natspec** comments at the start of it which helps us understand the working of the function.

6. Summary of the audit

Overall the code is clear on what it's supposed to do for each function.

My final recommendation would be to pay attention to adding **natspec** comments for anyone reading the code.

This is a very secure contract where users can safely participate in the **Crowdsale** of **PAY** tokens and use it to transfer around the **PAY** tokens.