Seugnhyuk Baek

Sbaek44@vt.edu

ECE5480 Fall 2018 HW #2.

1. Humans are said to be the weakest link in any security system. Give an example for each of the following:

(a) A situation in which human failure could lead to a compromise of encrypted data

A person used public computer in a library then left without log out securely. Next person who uses the computer can access all emails and browser data.

(b) A situation in which human failure could lead to a compromise of identification and Authentication

A malicious email sent from a hacker asking user id and password for a certain website. If a person believes the email is from legit administrator and gives information, then the hacker can access the website as if he or she is a valid user.

(c) A situation in which human failure could lead to a compromise of access control
In 2017, North American casino was hacked by a hacker. The hacker could get into casino's main system by IoT thermostat device hooked in fish tank.

2. Compare and contrast symmetric encryption with public-key encryption, including the strengths and weaknesses of each

Symmetric Encryption: Data between legit sender and receiver is encrypted using shared secrete key between them. This key is used to both encrypt and decrypt the data. This is very fast way to deliver encrypted message. However, if more actors (as sender or receiver) involve in the network, the secrete key will be distributed to larger population, leading higher chance of secrete key compromising.

Public-Key Encryption: In Public-key encryption, there are two keys to encrypt and decrypt a message. The receiver broadcast public-key to public. This public key is used to encrypt message which will be sent to the broadcaster. When the broadcaster receives a message from subscribers, the message is descripted using the private key that is designed to decrypt a message encrypted by the public-key. This method is slower then symmetric encryption but less compromising even the network becomes larger.

3.

Introduction

This is summary of short blog journal by Taylor Hornby titled “Salted Password Hashing - Doing it Right” retrieved from his web site CrackStation (https://crackstation.net/hashing-security.htm). The journal discusses salted password hashing to increase security of a web-based application.

Description

The author mainly discusses about how to store passwords in a web application. The main goal of this journal is not to build a noble system that theoretically impossible to bleach. Rather, the author discusses how to avoid password comptonization by simple yet powerful encrypted methods. It is easily assumed that not all users keep best practice for IT security. Users’ password and user id may be simply easy to guess so that a hacker can find very easily on dictionary; or too short so that the brute force attack leads to bleach.

While a user contributes to large portion in authentication failure with easy-to-guess passwords or accidently giving up his or her own passwords, the author insists that the application system must prepared for attack that already successful in other web applications. In such case, a hacker might have required user id and corresponding password. If a web application stores password in plain text format, bleaching of one system will lead to several other successful attacks. Therefore, passwords must be encrypted with safe hashing functions. If all passwords are stored as hash, hacker would have hard time to derive real password out of hashed text. However, this hashing technique can be vulnerable if a hacker already prepared with look up table. There are popular hashing algorithms open to public even to hackers. This means that a hacker can create fast look up table with existing known passwords. The hacker only needs to do to match hashes from bleached data to look up table. To eliminate this possibility, salt is added to password before it is processed by hashing algorithms. Each user granted salt or extra meaningless string that is attached to the user’s password then hashed with algorithm. This method will introduce randomness to hashed password make hard to compute original password from hashed password.

Analysis

The authors main point of this paper is to reduce malicious hacking by storing user passwords not in typical or easy to compute format. This method is not effective on attacks that uses valid user id and password which the hacker may acquired by phishing. The journal assumes hackers already posses basic look up table so that they don’t have to conduct brute force attack. If passwords are stored in database as hashed text with salt, it will be much hard to match user id and password just by seeking look up table. Overall, adaptation of salted hashing algorithm can make internet environment much safer. For example, company A’s database may be bleached, and user id and passwords are stolen. In this case, if all passwords are hashed, the hacker cannot plan for further attack using acquired information.

Conclusion

A user uses hundreds of web-based application or technologies in these days and most of them require to create an account to use their service. Therefore, most users use one or two user id and passwords to make their life simple. This means that comptonization of one system can lead to further IT security disaster creating millions of victims. The author gave a noble solution to avoid such catastrophe.