Emerging Technologies

Security Tool for increasing Security of stored user Data.

Introduction

I propose a tool to be used with android developer or other similar mobile development platforms that will focus on securing the storage of user data and passwords with better practices so that there is much lower chances of comprises to the Users identities and passwords. Some, especially newer, developers may not have experience or adequate knowledge to put in correct measures. The idea is it would be a plugin module that when sensitive data was being sent to a database for storage while still being light weight for running on mobile devices. It could also be used if the data was stored locally among IoTs so that if there was a compromise sensitive user data wouldn’t be lost. The idea is to assume a worse case scenario and try to protect user’s privacy even if data was compromised.

The plugin to the developer tool would suggest appropriate methods of password hashing and salting when storing a designated password variable type. It would come with some pre-set hashes such as MD5 or SHA1 to work with but allow the creation of user created hashes to be used on further applications. It would make all the hashing and salting automatic while making coding additions as small as possible to make the development process as streamline as possible.

Password hashing

Most applications such as email providers and social networking cites create an account for its user and to authenticate said user, they normally take a username and a password to allow them to log into the account. In principle most username-password pairs are stored in some sort of database. When a user logs in their credentials are matched against the databases stored ones. Though plaintext requires no additional work and would remain the quickest method it is undesirable should a major security breach occur the hackers would have every single username-password pair.

However just storing hash values of passwords does not offer protection against some attacks. An attacker can use a rainbow table which is a precomputed list of common passwords and their hashes which the hackers can compare to stolen passwords. The complexity of this attack is about 2^t/2 wherein the passwords file containing (2^t/2)t-bit has values and requires a precomputed database of 2^t/2 hash values randomly chosen passwords to find at least one password with a probability of at least 50% due to the birthday paradox[1].

A dictionary attack is a special kind of birthday attack where the passwords that are likely to succeed such as dictionary words are hashed as part of the precomputation of the attack. The generic complexity of a dictionary attacks is at most to that of birthday attack but often much less than this complexity with low success rate. These both are only efficient when their area a large amount of entries of passwords [1].

Password Salting

Password salting is adding a random string of characters to passwords before their hash is calculated to make password hashing more secure and it makes them difficult to reverse. The random string of characters can be combination of letters, numbers and other characters [2]. See appendix for a salting process which generates salt bits for specific binary string.

Refences

[1]

[2]

Appendix