CS 5435 – Security and Privacy Concepts in the Wild

Homework 1

Name: Shu-Wei Hsu(sh2575) *Spring 2024*

Exercise 3.1.1
For an offline brute-force guessing attack, the number of common passwords the attacker possesses would limit the number of guesses made by an attacker.
For an online brute-force guessing attack, not only the number of common passwords the attacker possesses would limit the number of guesses made by an attacker, but the number of times that the website tolerates inputting wrong passwords would also limit the number of guessing by attackers.
Exercise 3.1.2

The best strategy is trying q most probable passwords from the distribution. The probability of success is $1 - P(fail) = 1 - (1 - p(w_1))(1 - p(w_2))...(1 - p(w_q))$, where w_i is the i-th most probable password we can get from the distribution.

Exercise 3.1.3
Suppose $p(w_1) = 0.01$ and $p(w_k) = 2^{-20}$ for $k = 1, 2, ..., 1000000$. In this case the Shannon Entropy is 19, which is high. And the q-success probability is 0.01, which means the probability of cracking the password is 1%, which is pretty high.
In this example, we can see Shannon Entropy is not a good metric to measure password strength.

Exercise 3.2
Pepper is a value that is added after password before hashing. It's similar to salt but the major different is that the pepper will not be stored in the database along with users. It is often hard-coded or be provided as an environment variable. Therefore, all users will be sharing the same pepper.
The power of pepper is to add another layer of security. Even if the user database is breached. The attack still need to crack the pepper to do malicious things.

Exercise 3.3
I would add Account Lockout Policies. If the number of times of inputting wrong password exceed certain number, I would send an email to notify the user(assume we store email in the database) to change the password.

Exercise 3.4
The first drawback is that this policy offers bad user experience. User would get frustrated if they need to comply with too many rules.
Another drawback is that it makes harder to memorize the password as an user.
I would suggest adding password strength meter for users to see if their passwords are strong enough. The password strength is calculated by deep learning model and using the concept of "Guess Rank". It could nudge user to use stronger password.