Preliminary Research

In this preliminary research, we will be looking into how passwords are stored online, and how their leak can differ in severity depending on how the information was saved by the service provider.

There are several methods for a website or a service provider to store user passwords these days. Some are outdated and dangerous while others are becoming the standard by the security point of view. Obviously, letting large and reputable service providers handle user login/security information, through Google or Facebook login for example, is the best way to avoid a leak. However, this is not always available or best fit for projects nor is it readily available for us to test with. Thus, this method will not be considered for this project.

Below is the list of methods that will be tested/replicated:

- Bad: Storing password (and possibly its hint) as a plain text in a database. This is bad since in case of a data breach, all passwords along with their user info will be read easily by the attackers.
- Bad: Hiding passwords behind an encryption. This is only marginally better than above since all passwords are only one key away from being read by attackers. If there are insiders providing encryption key, it is nearly useless as a security method.
- Better: Hashing every password. This is a better method but has its own flaws. If there are multiple repeating hashcode, this can be a giveaway for a common password. Attacker can also use something like rainbow-table that replace computing power with disk space to accelerate the cracking process.
- Best: Hashing with salting. Hashing with slating is a method where during the hashing process of the passwords, the service provider adds random string of characters or numbers and hashs the combined product of salt and the actual password. This way, during the check, password can be recombined with the salt to compare against the hashcode. This provides very high security for every single password's hashcode will be different from each other and cracking of the code itself can be near impossible depending on the implementation.

References:

https://www.2brightsparks.com/resources/articles/introduction-to-hashing-and-its-uses.html

https://www.youtube.com/watch?v=8ZtInClXe1Q