Thoams Hart
CS 465 HW #9

Three things I learned from *An Administrator's Guide to Internet Password Research* were that some sites still store plaintext passwords or don't store them properly, rainbow tables exist to guess hashed passwords, and the password requirements most sites give (like having a certain number of characters and a combination of numbers, characters, and/or symbols) don't improve password guess resistance.

The article said, "Recent large-scale breaches have provided significant collections of plaintext passwords, allowing study of actual user choices." This allowed them to do some studies on the nature of peoples' password choices, but it surprised me that people could have the responsibility of storing passwords without knowing they should be hashed and salted. Of the eight companies the article lists, only two of them have properly hashed and salted the passwords, while the others have only hashed them, encrypted the passwords to a reversible state, or simply stored the plaintext. These are well-known companies, ranging from 1-150 million accounts on each site. That is a ton of insecure passwords. For one example, 90% of LinkedIn's passwords were guessed in six days because it was hashed but not salted. The recorded times are from about 10 years ago, so maybe they've resolved those issues.

Rainbow tables allow attackers to guess hashes with the assumption that passwords fit into a finite dictionary. Attackers find the hash of each word in that list and store the plaintext and hash together. They can also reduce the storage this take by using "repeatable sequences of password hashes" called chains. (I believe this was the method used to crack 90% of LinkedIn's passwords.) There are additional services to construct these tables, making the process easier. The article emphasizes that offline attacks using these tables are useless if passwords use proper salting. You must have leaked password hashes to break them.

Password requirements are apparently a naïve approach to resisting password guessing attempts. Some require special characters, a capitalized letter, numbers, and a certain length. As a rule of thumb, users are also urged to make their password random sequences, but the entropy does little to defend guessing attempts either. The article discusses a study on these requirements. 89% of passwords that required capital letter either had one capital letter at the beginning or they were all capital. This doubled the amount of guesses an attacker needed to make, but that is nothing in computing terms. 14% of these passwords were cracked in 50,000 guesses or less. Only 7% were cracked in 50,000 guesses if a special character was required, but this is still not huge. With passwords that required all the previously mentioned requirements but had varying password lengths, there wasn't much variation in the number of guesses it took to crack them below 10^6 guesses. Meters that indicate the strength of a password also use flawed logic and are not good estimates for password security either. Overall, a person should try to have a complicated, long password to an extensive degree on sites that need greater protection. All sites will rely heavily upon hashing and salting passwords properly for proper security.

My first question is: How are there not more password break-ins if they seem so easily breakable in this article? My second question is: Have most companies realized the flaws of storing passwords improperly, or are my passwords still at great risk without being hashed and salted properly?