Password Strengthening Lab

Team Members: Edwin Lopez, Arnel Soriano, Matthew Teng

Our password strengthening program is designed to enhance the security of passwords by transforming their structures into different ones by manipulating a maximum of 3 characters. This process is grounded on the idea that more common passwords and password structures are cracked earlier in brute force attacks. This is due to the way that there are characters and character patterns which are more likely to appear than others in any given password.

Password Strengthening Method

The password strengthening method processes character and structure frequencies from rockyou.txt.6.4.a.pcfgc, focusing on how often each character appears at specific positions and the frequency of different password structures (composed of uppercase, lowercase, digits, and symbols). This analysis helps identify common usage patterns.

For strengthening, each password is analyzed to find a less common yet achievable structure within three modifications, using both existing and potential new structures from the dataset. The strengthenPassword function then transforms the password's structure, selecting the least common character of the required type for each position. This is done while limiting changes to three, aiming to enhance strength without sacrificing familiarity.

The approach maximizes the effect of these three changes, either by altering the structure or by substituting characters, thereby increasing security and reducing predictability, making the password more resistant to common attack methods.

Things to Improve

The strengthening algorithm could use additional improvements in that we were unable to meet the 90th percentile 10^{12} guess number goal. It's hard to say what would bring the strengthening algorithm to that point without having done it. Before running out of time, an idea was to also use the character frequency values provided to not only turn common password structures into less common ones, but also turn them into completely new structures not found in the training password set.

The use of the character frequency for generating potential non-existent structures would greatly increase the strength of our passwords and help with runtime. Due to runtime constraints we had to cut back on generating non-existent structures. Our approach used brute force that repeated for every single password. This was not viable. A great way to increase our password security would be by using the character frequency to our advantage in order to find non-existent password structures strategically. We predict this would be a huge improvement.

Conclusion our implemented password strengthening method enhances password security by moving away from predictable structures and frequently used characters, making passwords less susceptible to common password attacks.