Making Secure Easy-to-Remember Passwords

High Level Outline

Philip Braunstein

My project seeks to compare three methods for making passwords:

1. Random string of numbers, lower and upper case letters, and symbols
    a. E.g.: ti$@98e(AkLM.O#
2. Passwords formed from the first letters of a song lyric or a phrase plus a number.
    a. E.g.: icgns12 (from: **I** **c**an't **g**et **n**o **s**atisfaction)
3. Passwords formed by a mental image of three to four all lower-case normal words as described in xkcd 936 (https://xkcd.com/936/)
    a. E.g.: correcthorsebatterystaple

Passwords will be evaluated on three qualities:

1. Bits of entropy
2. How easy they are to break
3. How easy they are to remember

The number of bits of entropy is trivial to calculate.

To determine how easy each of these are to break requires a password breaker. I plan to use John the Ripper on default settings to see if and how long it takes to break passwords formed by each of these methods.

In order to test how easy each of these to remember, I plan to recruit 5-10 volunteers from Halligan (or elsewhere). I will have each of them come up with their own password according to one of the methods described above. Then I will have them wait a certain amount of time (5 minutes should be sufficient). After this time has elapsed, the subject will be asked to write the password down as best as s/he can remember. I will come up with a good way to evaluate how close they are to remembering their password correctly. Percent of positions remembered correctly may be sufficient; however, I anticipate that something more complicated might be necessary (for example, what if someone gets a password entirely correct aside from forgetting one letter so that all of the other letters are shifted in position).

At the end of my project, I intend to either give a recommendation of which of these methods are best, or which further tests are necessary to determine recommend a method.