Further to my other article about passwords , here is an update on what constitutes a good password:

Good Users and Bad Passwords

Here's an extract:

What makes a strong password?

The strength of a password is typically described using the term password entropy, which is a measure of its **randomness**. It's not so much a measure of that specific password, as of all the *possible passwords* which contain the same range of characters (i.e. all the possibilities a computer would have to try in order to crack it by brute-force).

Entropy is usually expressed in bits: if we refer to a password as having *n* bits of entropy, it means that the entropy value is 2 to the power *n*. A single lower-case English letter has approximately 4.7 bits of entropy, because $2^{4.7}$ is approximately 26. So if a password only contains lower-case letters, then each will add another 4.7 bits of entropy (i.e. a two-letter password will have 9.4 and so on).

If we replace one or more letters with other characters, then the range (and therefore the entropy) will increase. There are 94 non-diacritic letters, numbers and special characters in US ASCII, so each will have approximately 6.55 bits of entropy (because $2^{6.55}$ is approximately 94).

Therefore an eight-letter password which might contain any of these characters will have approximately 52.4 bits of entropy, whereas a password of the same length with only lower-case letters will have 37.6 bits of entropy.

However a *sixteen-letter* password with only lower-case letters will have 75.2 bits of entropy.

To put that into some kind of context: a password with 52.4 bits of entropy might be cracked

by a desktop PC
in less than half an hour, while a password with 75.2 bits of entropy could take several hundred years. The longer the password, the more time it takes to crack, exponentially.

So in general terms, **a long password with nothing but lower-case letters is better than a short password with a mixture of characters** .