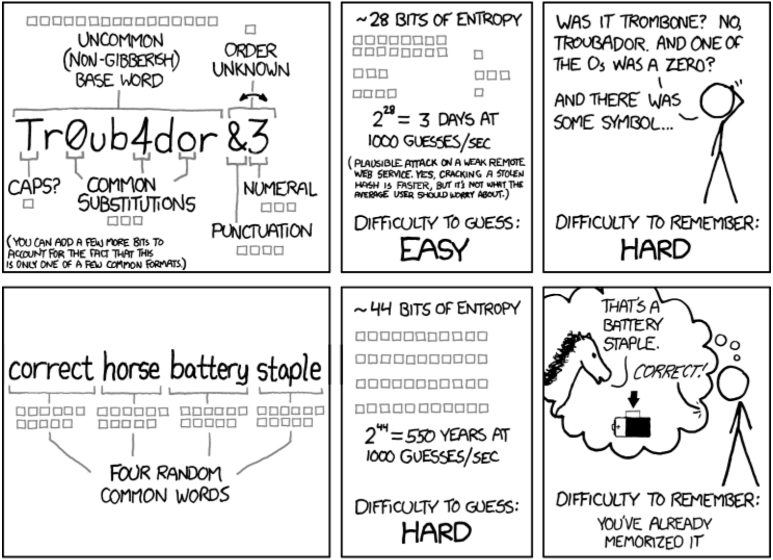
password1


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT

THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Linear congruential generator

$$S_n = (aS_{n-1} + c)\ \%\ m$$

Or

$$F(x) = ax + c\ \%\ m$$
$$S_n = F(S_{n-1})$$

Shannon Entropy

$$H(X) = -\sum_{x \in X} P(X = x) \log p(X = x)$$

When X represents a uniform random choice from a set,

$$p(x_i) = \frac{1}{|X|}$$

$$H(X) = \log |X|$$

Hashing cryptographically secure random number generator

$$\begin{aligned} S_i &= F(S_{i-1} \| E_i) \\ R_i &= F'(S_i) \end{aligned}$$

Where E_i is whatever entropy we have at the time of generation.
What this usually looks like when we have some entropy x to add:

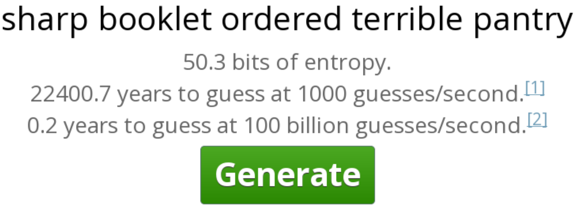
$$S_i = F(S_{i-1} \| x)$$

Given some hash function F_0, F and F' can be derived as

$$\begin{aligned} F(x) &= F_0(0 \| x) \\ F'(x) &= F_0(1 \| x) \end{aligned}$$

Mouseware

`http://www.fusionbox.com/mouseware/`



sharp booklet ordered terrible pantry

50.3 bits of entropy.

22400.7 years to guess at 1000 guesses/second.[1]

0.2 years to guess at 100 billion guesses/second.[2]

♡ HTML5

Fork me

`http://www.github.com/fusionbox/mouseware`