Christopher LaJon Morgan
CS 465
November 21, 2013

Password Cracker

Experiment:

Experiment	c/s	Time to Crack Lower case	Time to Crack Lower/Upper	Time to Crack 1!.aA
Weak	1898Kc/s * 4	0:00:00:11	0:00:05:43	0:00:22:40
Medium	"	n/R	n/R	n/R
Good	"	n/R	n/R	n/R
Best	"	n/a	n/a	n/R

1. Time Estimations:

Password Type	Possible Passwords	Cracks/Sec	Est. Time to Crack
6-Char Alphanumeric	2176782336	7592000	4.7 Min
8-"	2.82111E+12	7592000	4 Days
10-"	3.65616E+15	7592000	15 Years
12-"	4.73838E+18	7592000	19790 Years

2. I think Microsoft's password strength indicator is kind of close. If my little computer can break a weak password its pretty weak. Also, my computer has more and more trouble cracking a password as its Microsoft password strength increases. However, I would imagine that the scale should be a little bit more skewed, or articulate. Like weak should be 'you're only keeping out the honest', medium to good should be 'you might keep out a college student', and best 'really the only kind of good one'.

3.

Password Type	Possible Passwords	Cracks/Sec	Est. Time to Crack
6-Char Alphanumeric	2176782336	33100000000	0.06 Sec
8-"	2.82111E+12	33100000000	1.4 Min
10-"	3.65616E+15	33100000000	.5 Day
12-"	4.73838E+18	33100000000	11 Years

I think this just confirms my answer in part 2. I have friend that have several GPUs in which they use for heavy computing. So I was already thinking along these lines when I answered number 2. Really, with an attacker having a fast piece of hardware, one really might as well have no password at all, if the password is small and unsophisticated.

4. Using SHA-512 would improve password security. It would improve security because it is believed to be harder to perform a pre-image attack on SHA-512 than with MD5. This improves security because if something hashes to ones password it is believed to be the correct password. In other words, if it hashes correctly the attacker is in. Therefore, if when using SHA-512 it is harder to find something that matches to the hash, it has better security. Second, SHA-512 would also help to improve security because it runs slower than MD5. This improves security because it means that an attacker cannot perform as many cracks per second if SHA-512 is used.

5. Yes, the use of a salt does improve security because when a salt is used the hacker must appended each possible salt to a single guessed password and then hash each of the salted passwords. This means that a hacker cannot just guess a password hash it and then check it against each hash in a passwords file. Instead a hash must be calculated for each separate password. This increases security because it increases the computation time to crack a password.

6. No, if anything it would seem to increase its importance. Given that online attacks seem infeasible, the focus moves to something that is. Given that many systems have been compromised and password files acquired, it seems that protecting against offline attacks is still very import.

7. I think that it is totally foreseeable that in the somewhat near feature that hardware will exist that will render passwords with manageable lengths unusable. This will open up any system or process that has been protected by passwords. Meaning that anything involving password authentication will be vulnerable. We may have to move to biological identification or maybe physical token authentication, or some other method. All in all this might be the turning point that causes passwords to be replaced by what used to be a less favorable alternative.