**PROJECT 7 - PASSWORD CRACKING**

**Thought Questions**

**1. Assuming that you used your setup for this lab alone, how long do you calculate that it would take to crack a 6-character alphanumeric password? 8-characters? 10-characters? 12-characters? (use the c/s measurement from your experiments).**

During my experiments, it took on average about 3 minutes to crack two character passwords and about 40 minutes for four character passwords with an average rate of 280000 (meaning a two character password took 50.4 million attempts to guess correctly and a four character password took 672 million attempts). Assuming that correct guesses follow an exponential curve (calculated using y = Pert as explained at http://www.algebra.com/algebra/homework/Equations.faq.question.117703.html):

A 6-character password would take 8.96 billion guesses (8.89 hours with my current setup)

A 8-character password would take 119.47 billion guesses (4.94 days with my current setup)

A 10-character password would take 1.5928889e+12 guesses (66 days with my current setup)

A 12-character password would take 2.1238519e+13 guesses (878 days with my current setup)

**2. Do you think that the Microsoft password checker is a good indication of actual password security? From the results of your experiment, what is your recommendation for minimum password length? Be creative in your response. Imagine what hardware and resources a potential attacker might have, and briefly justify your assessment of the attacker’s capabilities.**

I do not think the Microsoft password checker is a good indication of actual password security. Based on my experiment, I would recommend a password of at least 12 characters that includes punctuation and special characters (so that the password won't be cracked by a dictionary attack). Recently there have been reports on the internet of GPUs that can crack "any" 8 character password in under 6 hours, so to be secure you need a longer password. Because security essential increases exponentially in relation to characters of the password, an additional 4 characters would significantly increase the cracking time from hours to months.

**3. Recently, high-end GPUs have revolutionized password cracking. One tool, ighashgpu, is able to perform 1.3 billion MD5 hashes per second on an AMD Radeon 5850 (a 2-year-old, mid-to-high range video card). Whitepixel, another tool, claims that it can perform 33.1 billion hashes per second using 4 Radeon 5970s. Consider your calculations in question #1, and redo them assuming you had access to a system with 4 Radeon 5970s. Do your answers for question #2 change?**

A 6-character password (8.96 billion guesses) would take

0.27 seconds with access to a system with 4 Radeon 5970s

A 8-character password (119.47 billion guesses) would take

3.6 seconds with access to a system with 4 Radeon 5970s

A 10-character password (1.5928889e+12 guesses) would take

48 seconds with access to a system with 4 Radeon 5970s

A 12-character password (2.1238519e+13 guesses) would take

10.7 minutes with access to a system with 4 Radeon 5970s

I think my estimation for password length might have been a bit short for question #2.

**4. Fedora 14 and other modern Linux distributions use a SHA-512 (rather than MD5) for hashing passwords. Does the use of this hashing algorithm improve password security in some way? Why or why not?**

Yes, SHA-512 is an improvement over MD5 because MD5 has been proven to no longer be collision resistant and is considered "broken." SHA-512 hashing is still collision resistant and much harder to crack.

**5. Does the use of a salt increase password security? Why or why not?**

Salting passwords renders lookup tables, reverse lookup tables, and rainbow tables useless in password cracking, thus improving password security. Also, even if a hacker cracks one password in a database, they won't be able to crack the other passwords because they all use different salts.

**6. Against any competent system, an online attack of this nature would not be possible due to network lag, timeouts, and other security. Does this knowledge lessen the importance of offline password attack protection? (Hint: Think about the recent breaches of the PlayStation Network, Steam, and others)**

This does not lesson the importance of offline password attack protection at all. If attackers can download a database of password hashes, they can perform the attacks offline until they get the correct hash, and then do the attack on the first try online.

I definitely see the need for minimum password lengths to be extended in the near future. I think the biggest vulnerability that will arise is users choosing easily remembered (and thus, easily cracked passwords) because it is much harder for users to remember long, complex passwords. So if they have to make a long password, users will probably make them easy to remember.

The next in password evolution should be a two-step password process (sort of like Google's two-step verification) that would require users to not only have a typed password, but also another key. This key could be a randomly generated hash specific to them, an NFC chip that they physically possess, or anything else that could only be replicated with physical access simultaneous with digital access.