Isai Mercado Oliveros
misaie
November 16, 2015

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).

6 characters takes 21 seconds
8 characters takes 7 hours
10 characters takes too long
12 characters takes too long

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 think that if people do not know anything about cracking passwords, a password checker is good, but according to my experiments, and some information found online the best defense is to make the password very long and use all the possible symbols. My minimum password length is 10 characters.

Attackers vary depending on the objective. If the target is a normal person, possibly the attacker would be a normal person with a pc as powerful as the target; on the other hand, if the target is the FBI, then the attacker might have a cluster of servers trying to crack a password.

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?

Yes, it changes because now the minimum password length would need to be 12 characters
6 characters takes 1 second
8 characters takes 1 minute
10 characters takes 1 day
12 characters takes too long

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?

I think it does not help a lot because attackers still have lists of most commonly used passwords, and password crack software like john the ripper brake the password by grabbing words from lists and playing with the word around by using some rules, so using a more secure hashing function does not help a lot because people still use common words or short words with lowercase letters and numbers, but they do not add capital letters or other symbols.

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

A salt increases security by preventing the creation of rainbow tables for a particular hash function, so that the password crack software has to brute force the attack but it does add protection to weak passwords

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)

It still important to know because users need to understand how to make passwords. For example they need need different passwords for every site. They have to avoid using the common passwords. The best protection is to have long passwords, with the most symbols possible, such as capital letters, numbers, and extra characters. In addition, it is important to know in order to have a good user password storage system. For example, using salts in addition to user passwords stops attacker from making rainbow tables for the hash functions, and