The Advent of Uncrackable Passwords (2nd Edition)

Base character set for the English language (as taken from LC5's Dictionary):

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/

Total (T_B): 68

LC5 operates by taking the two halves of the Windows (NT)LM passwords, the first 7 letters, and the second 7 letters. LM passwords cannot exceed this 14 character limit, and 98%+ of NTLM systems can revert to LM: which means that the additional security provided by NTLM is useless. Password cracking machines perform the crack on both halves of the password, thus reducing the number of characters from 14 to 7.

Total Permutations (P_B): $68 + (68)^2 + (68)^3 + (68)^4 + (68)^5 + (68)^6 + (68)^7$

Explanation: There are 68 possible values to fill in the first number, 68 for the second, etc. You can have a one character password, a two character password, and so on. A two character password 'pp' differs in its hash from the one character password 'p' So you add up the total combinations of one character passwords and the two character passwords… all the way up to the 68 character passwords.

Upon the introduction of the extended non-ascii keymap to the character permutation sequence, **at least** another 256 characters will be added to the character set above.

Total (T_E): $68 + 256 = 324$

Total Permutations (P_E): $324 + (324)^2 + (324)^3 + (324)^4 + (324)^5 + (324)^6 + (324)^7$

Ratio (P_E)/(P_B): 55101.2

In other words, it will take approximately 5,501 times longer to crack! All this, without the introduction of any more than 256 characters (the original non-ascii but still Unicode keymap, or the introduction of another languages character set (think: Chinese!)).

The number-crunching machine I use for memory/CPU intensive operations is a 3.25 GHz P4 with Hyper-Threading Technology. It has 512x2 Dual-DDR400 megabytes of memory; and runs Windows Server 2003 for stability. LC5 can be considered dependant only on core CPU performance. On this test machine, LC5 takes up to 54% of the CPU process, and operates with a keyrate of about 4449740 passwords per second.

Time for Base Set: 1533422.6 seconds or 425.95 hours (17.7 days).

Time for Extended Set: 84493426690 seconds or 23470397 hours (977933.2 days)!!!

It is technically impossible for **anyone** to take this long to crack a password. No private company would have either time or resources, and if the governments were to try, wars would break out and countries would dissolve and alliances will have reformed before a single password can have been cracked, even with superior computing capabilities.

The only consolation is that it can run multiple passwords at once, so that if you have a bank of 10,000 passwords to be cracked, it will take just as long to crack as if you were trying only one. However, this isn't much of a consolation, since it is still just as impossible. *Unless* you encrypt the password hashes with salts, then the passwords can be hacked **only one at a time.** This is of such importance that it is a wonder that Microsoft has made it incredibly hard for us to enabling salting on our PCs, but upon enabling of salting Windows will automatically add a random piece of data to the end of each password *then* follow through with encryption. This results in each password following a different encryption function, and makes simultaneous brute-force cracking impossible.

Why this works and nothing else does:

Why should this method work in making passwords "uncrackable" when thousands of other methods in the past have failed? Here the true usefulness of non-ascii is revealed. What makes this method the best is that there is nothing to lose, there is no trade-off of security or using one method instead of the other. These characters can be used on all systems, Windows, Macintosh, Linux, Unix, Linspire, etc. and will work equally powerfully on them all: it is an OS **independent** solution.

What's more, using non-ascii passwords does not have to make up your sole password security system. It **can be** but **is not necessarily** a stand-alone solution. You may use it as it is presented above and secure your PC with a password purely of the likes of =: ╟ʖ±衫 or you can integrate this form of security into **any other security system you use now**. You can lock PGP secure codes with non-ascii passwords as you can sign Verisign certificates with it. You can take these letters and convert them to binary and use them as codes or you can apply complex algorithms that will leave the world stunned and in the dark regarding what you are conferring about. This code, in short, can be the **most powerful complement** you will ever add to your existing system, making your passwords **at least** sixty thousand times stronger!

Of course, the machine I have at home pales significantly upon comparison to the hundreds or thousands of grid networked CPUs the NSA or Interpol will use upon attempting to crack a password, and since the security of the above method relies solely on the amount of time and resources such a crack would take, and both of these can be circumvented under these circumstances. However, did anyone stop to think about how the NSA cracks passwords and how they know what to crack. A study undertaken by the *Guardian* in March 2005 looked into the US and British intelligence. It shows that in all likelihood passwords are collected at random, whatever is come across, and fed into machines hose sole purpose is to crack them.

While this is possible when dealing with simple ascii-based passwords, which can be cracked within 30 seconds on their machines (an estimate) imagine trying to feed a list of passwords that use the non-ascii encryption specified above. It will take 1653036 seconds a password!! This exhausts the resources of governments and creates a back-log of passwords to be cracked. It consumes time, money, and energy, and can be fatal to a government trying to stop terrorists or a thief hacking into the CIA.

One more point to keep note of is that as of today throughout my research in security I have not heard or read a single reference to using non-ascii passwords such that the government or the likes thereof do not know of its existence (such that we can tell) and will **not** be attempting to include this non-ascii database of characters into their machines. Not a single cracking program makes mention of this or has a letterset that includes these characters. If the hacker doesn't know that this system was used (as is most likely) he or she will **never** recover the password.

It is; however, very important to keep in mind that this is the maximum time required for cracking a password of this sophistication, and also **assuming** that it is seven characters in length or more. **But** it does not matter how many (or even if any at all) of the extended character set letters are in use, so long as LC5 has been programmed to attempt them. And if it hasn't, then the password cracking will most certainly fail. A password of shorter length will take much shorter time to crack, and there is the luck factor: LC5 tries the passwords in brute force mode in a more-or-less random order. Through it, it may be that the very first seven character password attempted is the solution, but it was incredibly more likely that it isn't. Remember Murphy's Law: *If you are looking for something it will always be in the last place you look, no matter where you start.*

Good Luck!

-The Computer Guru
ComputerGuru@spymac.com