

Re: Cross platform password string encryption

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2006-08/msg01536.html>

- *From:* pegguru@xxxxxxxxx
 - *Date:* 26 Aug 2006 06:12:41 -0700
-

Ozzker wrote:

.... snip...

I don't see what programming languages or platforms have to do with it as long as they implement the same algorithms ..

anyway, regards,
Oz

Platforms: different microprocessors use different schemes for handling memory. For example, some processors are "big-endian": when they store a 16-bit integer (in two consecutive memory locations), they put the 8 most significant bits in the lower address and the least significant bits in the higher address – same thing if they write the 16-bit integer to disk. Other processors are "little-endian": they put the 8 least significant bits in the lower address and the most significant bits in the higher address – again, same thing if they write the 16-bit integer to disk. You can imagine what happens if a little-endian processor reads a file of integers created by a big-endian processor.

[NB: I may have gotten the "little-endian/big-endian" definitions backwards – I consumed way too much sugar last night and really need some programmer candy (aspirin).]

Languages: some languages are very specific about how they format and store various data types. Two such languages need not agree on how to format and store the same data type. And no language needs to be tied to a microprocessor's view of a data type (provided the user is willing to put up with additional processing). Bottom line is that if I write a program in language X and program the same algorithm in language Y, even running on the same computer, if I am not very careful in out data are read from/written to disk, the two programs may not be able to share data.

As computer science moves into more abstract concepts, I find that many young programmers have very little concept of what happens at the

Re: Cross platform password string encryption

microprocessor level.