Gregory E. Borter, Systems Coordinator of Silver Springs Alfa SmartParks, Inc., wrote to me with an interesting series of questions:

Message text written by "Gregory E. Borter"
>I've been reading about security problems with the various OS components, both Windows and Linux, and the problems with security with applications software. Where is the best place to start implementing system security?<

It seems to me that security should be integrated into the requirements analysis, the design stage for software, the operating system security kernel, in corporate policy development and in human awareness, training and education programs.

>Should security start with the computer programming languages themselves, or their support libraries?<

PASCAL uses strong typing and requires full definition of data structures, thus making it harder to access data and code outside the virtual machine defined for a given process. In contrast, C and C++ allow programmers to access any region of memory at any time the operating system permits it.

There are several sets of security utilities available for programmers; for example, RSA has a number of cryptographic toolkits. Some textbooks (e.g., Schneier's _Advanced Cryptography_) include CD-ROMs with sample code.

>Are there any computer languages that have security features built-in to the language itself?<

Not to my knowledge, but I'm not an expert in languages.

>With so many PCs linked via networks and the Internet, shouldn't all programs be coded with the assumption that the programs will be operating in an environment where they may very probably be subject to hostile attack?<

Yes, but the difficulty in testing for security is that there are so many possible ways to generate security holes in code. Buffer overflows, for example, are the most common form of security exploit, but clearly the programmers never thought to impose length restrictions on the input strings being handled by Web server software.

>Do any current computer programming languages give programmers tools with which to implement security best practices in their code?<

All computer languages allow you to write code as well as you can <smile>. I think that strongly-typed languages may offer better constraints on programmers, but the essential issue is that the programmers continue to think about security as they design and implement code. Java
does include provisions for limiting access to resources outside the "sandbox" reserved for a process, as described in the books by Felten and McGraw.

* * *

More on this subject in the next column.

* * *

References:


RSA Data Security < http://www.rsasecurity.com/products/ >


* * *

Mich Kabay can be reached by e-mail at <mkabay@atomictangerine.com>. He invites inquiries about a wide range of information security courses and INFOSEC consulting services that he and his colleagues at AtomicTangerine would be delighted to deliver to your employees at your site and at your convenience. For Web-based or CD-ROM online training in security from our INFOSEC University project, see < http://infosecu.com >.

For information about AtomicTangerine, visit < http://www.atomictangerine.com >.

Copyright © 2001 M. E. Kabay. All rights reserved.

Permission is hereby granted to Network World to distribute this article at will, to post it without limit on any Web site, and to republish it in any way they see fit.