In this series of articles, we’ve been looking at undocumented and unauthorized methods for gaining access to systems. In this article, I summarize some basic approaches to preventing back doors in source code. Network managers may not be directly involved in software quality assurance, but it would be a Good Thing to make sure that the quality assurance folks in your shop are aware of and implementing these principles before you install their software on production systems and networks.

* * *

Documentation standards are not merely desirable; they can make back doors difficult to include in production code. Deviations from such standards may alert a supervisor or colleague that all is not as it seems in a program. Using team programming (more than one programmer responsible for any given section of code) and walkthroughs (following execution through the code in detail) will also make secret functions very difficult to hide.

During code walkthroughs and other quality-assurance procedures, the search for back doors should include the following:

* Undocumented code
* Undocumented embedded alphanumerics
* Peculiar entry points
* Unexplained functions
* Code not executed during testing.

Every line of code in a program must make sense for the ostensible application. All alphanumerics in source code have to make sense; a more difficult problem is dealing with numeric codes which may have a hidden meaning. Every entry point for a compiled program must make sense in the programming context.

Every line of code must be exercised during system testing. Test-coverage (sometimes called “code coverage analysis”) monitors show which lines of source code have been executed during system tests. Such programs identify the percentage of code that is executed by a test or series of tests of programs written in a wide range of programming languages; however, each programming language may require its own test-coverage tool. The monitors usually identify which lines of source code correspond to the object code executed during the tests and which were left unexecuted. They can also count the number of times that each line is executed. Finally, test-coverage monitors may provide a detailed program trace showing the path taken at
each branch and conditional statement.

It would be nice if the major software vendors who provide operating systems and utilities were also aware of these principles. Certainly some of the quality-assurance teams at Microsoft must not have been applying such tools diligently in recent years. For example, in Excel 2000, you can activate a spy hunter game that uses DirectX for graphics <http://www.eeggs.com/items/8240.html>; in Excel 97 and later, you can load an unauthorized flight simulator <http://www.eeggs.com/items/29841.html>.

In the next and final article in this series, I’ll review methods for spotting remote administration Trojans (RATs) while they try to install themselves and finish with a note on getting rid of RATs.

***

For further reading:

Diane Levine’s chapter on software development and quality assurance in the _Computer Security Handbook, 4th edition_ (see shameless plug below) is an excellent primer on how quality assurance is fundamental to security.

Steve Cornett has an excellent paper on test-coverage monitors at <http://www.bullseye.com/coverage.html>.

Paterson Technology provide good information on their Web site <http://www.patersontech.com/TestCoverage/>

Testingfaqs.org has a wealth of materials about quality assurance and testing <http://www.testingfaqs.org/>

There’s an extensive list of test-coverage tools at <http://www.testingfaqs.org/t-eval.htm> with short descriptions of each tool and pointers to where they can be obtained.

***


M. E. Kabay, PhD, CISSP is Associate Professor in the Department of Computer Information Systems at Norwich University in Northfield, VT. Mich can be reached by e-mail at <mkbay@compuserve.com>. He invites inquiries about his information security and operations management courses and consulting services. Visit his Web site at <http://www.mekabay.com/index.htm> for papers and course materials on information technology, security and management.