The critical importance of integrating security into programming is obvious to anyone who thinks about it and has been the subject of countless minatory or sometimes pleading articles; Google “secure programming” as one example of appropriate keywords and you’ll find nearly a million hits. Back in 2001, I wrote five Network World Security Strategies columns on the subject which I later collected and updated as the short paper “Programming for Security” that’s currently on my Web site. Microsoft’s Michael Howard and Steve Lipner published Writing Secure Code, Second Edition and The Security Development Lifecycle; Michael Howard and David LeBlanc wrote Writing Secure Code for Windows Vista.

Now the National Security Agency, working through MITRE Corporation, SANS, and dozens of industry experts from many other organizations, has published a valuable list of the top 25 most dangerous programming errors. The best description of the project that I have found is the SANS Institute report. SANS provides a detailed summary of the issues, including this introduction:

Today [January 12, 2009] in Washington, DC, experts from more than 30 US and international cyber security organizations jointly released the consensus list of the 25 most dangerous programming errors that lead to security bugs and that enable cyber espionage and cyber crime. Shockingly, most of these errors are not well understood by programmers; their avoidance is not widely taught by computer science programs; and their presence is frequently not tested by organizations developing software for sale.

The impact of these errors is far reaching. Just two of them led to more than 1.5 million web site security breaches during 2008 - and those breaches cascaded onto the computers of people who visited those web sites, turning their computers into zombies.

SANS provides a list of the errors with a link from each to the MITRE database called the Common Weakness Enumeration (CWE). That site explains,
The list itself (<http://cwe.mitre.org/top25/index.html#Brief>) divides the errors into three major categories:

- Insecure interaction between components
- Risky resource management
- Porous defenses.

Readers will find the threat model (<http://cwe.mitre.org/top25/index.html#Appendix_B>) that was used in ranking the weaknesses particularly interesting. It presupposes a relatively skilled hacker intent on data theft or theft of resources and willing to invest at least 20 hours per target software package. The full process used in selecting the top 25 is documented (<http://cwe.mitre.org/top25/process.html>) and there’s also a list of 23 weaknesses that almost made it into the list (<http://cwe.mitre.org/top25/cusp.html>)

This research project will be enormously valuable to working programmers, instructors in computer science, computer engineering and information assurance programs, and students in those disciplines.

* * *

M. E. Kabay, PhD, CISSP-ISSMP (<mailto:mekabay@gmail.com>) specializes in security and operations management consulting services. CV online (<http://www.mekabay.com/cv/>)

Copyright © 2009 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.