Introducing "Insecure IT"

Rick Kuhn, *US National Institute of Standards and Technology*
Hart Rossman, *SAIC*
Simon Liu, *US National Library of Medicine*

Our goal, of course, is to offer ideas to improve IT security, both by looking at ways it can go wrong as well as by covering good practices. As most security practitioners and researchers have seen, new technology developments nearly always introduce a period in which attackers find relatively easy ways to exploit weaknesses, followed by a gradual closing of vulnerabilities. Wireless networking is a classic example—initially more than half of home users, and a high percentage of business users, installed 802.11 wireless with no security measures. Some spectacular incidents resulted from widespread ignorance of basic wireless security measures, such as cases where retailers operated online point-of-sale systems with unsecured wireless. By analyzing vulnerabilities in real systems, we hope to encourage readers to not only avoid similar problems in their own systems but possibly generalize the lessons to new technologies as they appear. Insecure IT will appear regularly in this magazine, so we encourage readers to submit articles and share their lessons with the world.

Understanding Vulnerabilities

In keeping with our theme of understanding vulnerabilities to improve enterprise security, we should first take a look at the current state. What are the trends in enterprise security, and where do we stand today? We can examine these questions in two ways: attacks and the vulnerabilities that attackers target. The latter bears directly on an organization's cost to protect its assets because it indicates the effort required to patch applications and close security holes as they're discovered. Using the US National Institute of Standard and Technology's (NIST's) National Vulnerability Database (NVD), we can get a sense of where we are today and what will be important in the near future. The NVD provides fine-grained search capabilities for all known vulnerabilities and is continuously updated to provide data for automated vulnerability management, security measurement, and compliance. With data going back to 1997, we can also use NVD to see trends in IT vulnerabilities over the years.

The NVD data in Figure 1 gives us some good news and some bad news. Clearly, vulnerabilities have increased dramatically in the past few years, and the increase has come from the most severe ones. But data for the past two years show a downward trend (2008 figures projected from 10 months of data). Although it often seems that software is full of holes and only getting worse, things really are improving.

Of course, this improvement is relative to the explosion of new vulnerabilities we've seen since 2003, and no one responsible for their organization's IT security can be happy with the appearance of more than 5,000 new vulnerabilities in a year. Nevertheless, this is the first two-year decline in the data, and the decline from 2007 to 2008 was much more dramatic than the previous year. It's also important to note that this chart covers data from thousands

 Published by the IEEE Computer Society 1520-9202/09/$25.00 © 2009 IEEE

of products. Digging in to the data a bit more, Figure 2 shows the types of vulnerabilities discovered in 2008. We categorized the vulnerabilities in Figure 2 by using the Common Weakness Enumeration (CWE), which defines a standardized description of software weaknesses designed to provide a common language for describing software security weaknesses. Using CWE, developers and analysts have a standard definition of terms for investigating security problems in architecture, design, and code. CWE also helps system administrators compare tools that attempt to find security weaknesses.

Buffer overflows, long the most common security bug, are now a distant third behind two Web-based vulnerabilities, SQL injection and cross-site scripting. As we can see on the left-hand side of Figure 2, traditional vulnerabilities affecting operating systems and standalone applications have become relatively rare. For example, the CWE found only 13 reports of race condition exploits (changing a file link between when the operating system checks the time permission and when the requested operation is performed). Careless applications of cryptography, such as employing a weak encryption scheme, used to be common as well, but the CWE found only 26 examples in 2008. Some old favorites, however, remain perennial problems, such as poorly configured access control and failure to validate input. Ultimately, it appears that software developers are finally beginning to turn a corner in their efforts to stamp out security-critical bugs, but the data in Figure 2 clearly show that newer technologies, such as Web services, bring new bugs to catch.

Implications

What this means for software developers and system administrators is that their vigilance is paying off across systems with a wide install base and significant time in the field. Although this past summer's announcement of a significant DNS flaw reminds us that core protocols and services still require additional scrutiny and research, it's clear that industry has adopted some of the lessons learned and that best practices have been proven out.

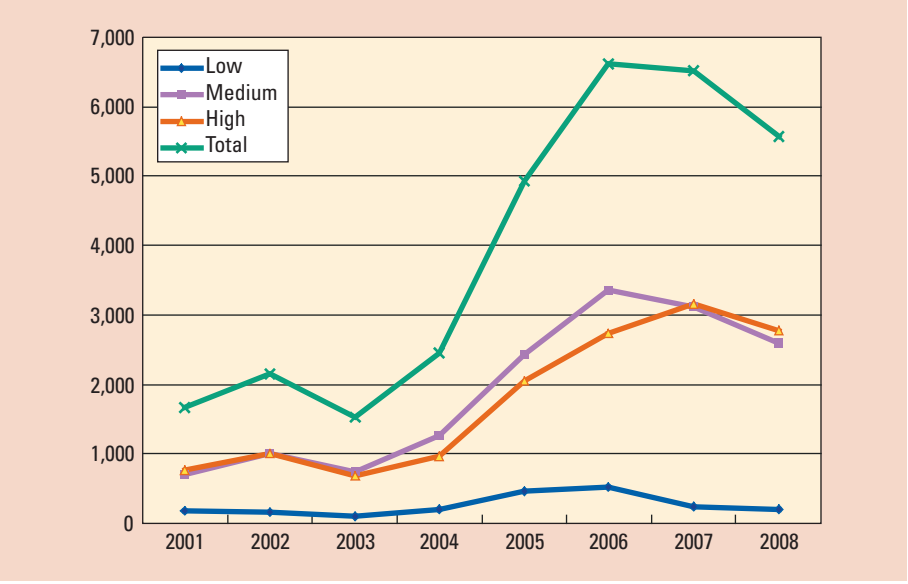

Figure 1. Vulnerabilities by severity. Vulnerabilities have increased in the past five years, but are starting to decline from a high in 2006.

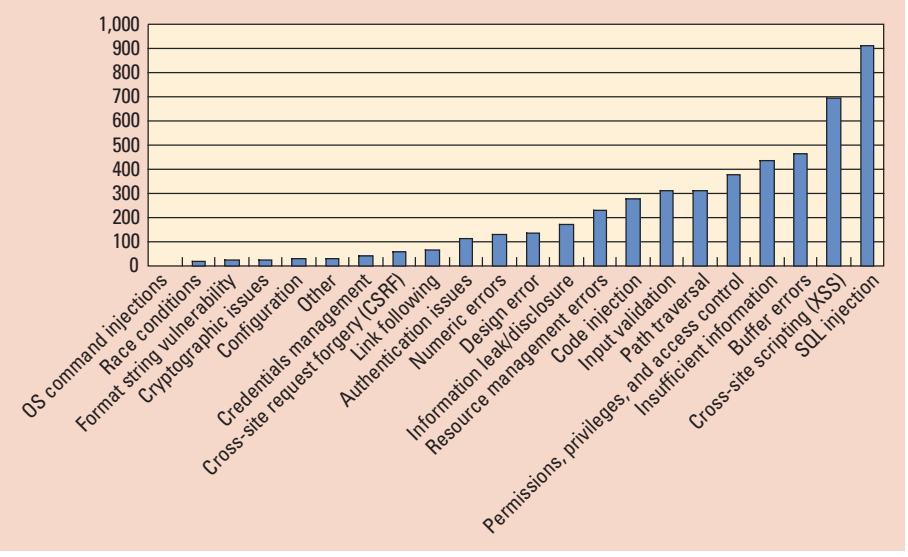

Figure 2. Vulnerabilities by type (January–October 2008). Today's most common vulnerabilities involve Web applications.

However, emerging technologies and new use cases for established systems are providing fertile ground for new types of vulnerabilities susceptible to an ever creative and persistent adversary. Priorities for attackers and defenders alike have moved to the application space, with an emphasis on anything Web-oriented or net-centric in nature. We can only expect this trend to accelerate with the proliferation of "always on" robust mobile computing platforms ranging from smart phones to netbooks, and the ever increasing prevalence of net-enabled consumer products in every aspect of our lives. The walls of the enterprise have become blurred, and software developers and system administrators will continue to experience an evolving landscape rife with opportunity to actively manage the risk of the systems they develop, deploy, and operate.

Help comes in a variety of forms, from community-driven organizations that promulgate best practices and vulnerability watchlists such as the Open Web Application Security Project (www.owasp.org) all the way to you, the reader. We heartily encourage your thoughts and look forward to including your submissions in future columns and as part of our upcoming annual issue focusing on security.

Acknowledgments

We're grateful to Chris Johnson at NIST for providing data from the National Vulnerability Database. As a disclaimer, certain software products are identified in this document. Such identification doesn't imply recommendation by NIST or other agencies of the US government, nor does it imply that the products identified are necessarily the best available for the purpose.

Rick Kuhn *is a computer scientist at the US National Institute of Standards and Technology. His research interests include information security, software assurance, and empirical studies of software failure. Kuhn has an MS in computer science from the University of Maryland, College Park. Contact him at kuhn@nist.gov*

Hart Rossman *is a vice president and CTO of SAIC. He also serves as a faculty member with the Institute for Applied Network Security. Rossman has a CISSP, a BA in communication from the University of Maryland, College Park, and an MBA from the University of Maryland, Robert H. Smith School of Business. Contact him at hart.m.rossman@saic.com.*

Simon Liu *is the director of information systems at the US National Library of Medicine. His research interests include IT architecture, cybersecurity, software engineering, and database and data mining. Liu has two doctoral degrees in computer science and higher education administration from George Washington University. Contact him at simon_liu@nlm.nih.gov.*