Michael A. Cusumano

Who is Liable for Bugs and Security Flaws in Software?

Attempting to determine fault and responsibility based on available evidence.

Last October, the *New York Times* published a story on a lawsuit in California being launched against Microsoft in the State Superior Court [3]. The suit, which is trying to get class-action status, claims Microsoft violated California consumer protection laws by "selling software riddled with security flaws." One concern is that, so far, software companies have protected themselves from product liability lawsuits by selling customers a license to use their software rather than the actual product and by requiring customers to sign off on a lengthy list of caveats and disclaimers. But this mode of licensing has come under fire as the computing world has become deluged with viruses that exploit flaws in software products.

When a reporter asked me to comment on this story, I found myself deliberating over several questions: One is whether or not software companies should indeed be held responsible for the quality of their products, like other companies. If General Motors or Ford sell faulty automobiles, you can be sure the courts will hold them liable when people or property are damaged. Why not software companies? Second, even if software companies are liable for product quality, can they realistically be expected to eliminate security flaws as well as more common defects? A third question is why has Microsoft become the focus of the latest lawsuit. Is the world's largest software company particularly bad at quality and security?

JASON SCHNEIDER

The potential damage to individuals and organizations from software defects is real and costly. The National Institute of Standards and Technology (NIST), for example, issued a study in July 2002 that claimed software quality cost the industry nearly $60 billion a year. Users bear about two-thirds of that cost, so this is clearly an issue for everyone [2].

First, as for whether software companies should be held accountable for product quality, the answer has always been yes, to a degree. The issue is really to what extent software companies are being held financially liable. Successful firms, including Microsoft, have learned how to respond to customer complaints and fix bugs in their products, either in special "point releases" or new versions. So far, though, no court of law seems to have found Microsoft legally liable for flawed software, and that is why there is a new lawsuit in California [3]. In other cases, however, customers have received large financial settlements from soft-

ware companies. For example, in February 2003, the manufacturing company Doskocil won $2.3 million in damages in an arbitration case against J.D. Edwards for shipping a "defective product" [4]. This case is not unusual. Almost every enterprise software company has faced customer lawsuits, and there are no doubt more on the way, with security flaws a rising issue. In fact, there is an entire industry of expert witnesses and law firms that deals primarily with customer liability claims and desires to hold software companies more responsible.

Most of the legal debates I know of center not around whether there is product liability, but around what is "acceptable" industry practice—such as for making delivery dates as promised or producing features that meet or do not meet original specifications in a customer contract, as well as for the type and levels of defects. The general philosophy held by software customers, software vendors, the American Arbitration Association, and the U.S. courts seems to be that software is a uniquely complex product that will probably always have some defects. But companies delivering software that exceeds the bounds of common industry practice are vulnerable to penalties. Because some software companies are much better than others at preventing, detecting, and fixing defects, it seems to me that many firms can do better and that courts should hold software firms more accountable for what they license or sell.

Most of the legal debates I know of center not around whether there is product liability, but around what is "acceptable" industry practice.

As for whether software companies can ever make their products error-free or invulnerable to security flaws, I think the answer is clear: no, they cannot. My own research suggests that companies have made remarkable progress in reducing defects, but they are still far from perfect. In a sample of 104 software projects from around the world, collected during 2002–2003, my colleagues and I found a median level of 0.15 defects per thousand lines of code as reported by customers in the 12 months after receiving a product.

In a smaller sample of 40 projects from the U.S. and Japan, reported on in 1990, we found a median level of about 0.6 defects [1]. But, despite the improvement, even 0.15 defects per thousand lines of code translates into 150 bugs for a million-line software program. Given that many software products today contain tens of millions of lines of code, it is no surprise to see hundreds of bugs in even the best-quality software from the best companies.

Moreover, when it comes to Web software, we have another problem: The Internet is an open system, like a mass-transit network in a city or like the world airline system. Sure, companies can make subways and airplanes more secure, and they are doing so, but at great cost and inconvenience to users. And, despite everyone's best efforts, a determined villain—a terrorist or hacker—will find a way into the system. So unless software developers and users go back to closed systems with very limited access, they will always have to deal with the inconveniences of criminal activity. Nonetheless, as the industry continues to establish more demanding norms for what are acceptable levels of quality and security, software companies will have to respond—on their own or under pressure from gov-

ernment and the courts.

Finally, when it comes to whether Microsoft's products are particularly riddled with bugs and other security flaws, I think not. Again, when products like Windows 2000 have upward of 30 million of lines of code, hundreds of bugs are inevitable. Microsoft is also the most vulnerable software company because Windows and Internet Explorer constitute a ubiquitous platform for the desktop, its server products are inexpensive and gaining in popularity, and the other flagship product, Office, has some 95% of the applications productivity market. But I also think Microsoft is still struggling to figure out how to design more secure software in an open, insecure world. Since Internet Explorer appeared in 1995, followed by an array of server products, Microsoft has no longer been developing software simply for the desktop PC.

A little bit of research confirms that security flaws are hardly a Microsoft-only problem. I did a search of approximately 100 articles appearing in *Computerword* on the subject during 2002–2003, and approximately half related to Microsoft products (various versions of Windows, Office, Passport, Windows Media, SQL Server, Commerce Server, and Internet Explorer). The other articles pointed out actual or potential flaws in a wide variety of commercial and open source products. Software developers have been busy fixing these problems, but the number and scope are still worrisome.

For example, nearly every version of Unix and Linux distributed by Sun Microsystems, IBM, and Red Hat, as well as Apple's Mac OS X server software, which is based on Unix, had security problems in how they transferred data between different systems. Linux also had problems in its compression library. Cisco's Internetworking Operating System (IOS) had a flaw in how it processed data packets that could lead to a hacker attack. Sun Microsystems reported flaws in its Java Virtual Machine that could allow hackers to take control of Web browsers and steal user identifications and passwords. Netscape reported similar problems with its browser as well as JavaScript. Sun also reported problems in its XDR library product and user authentication software. Oracle warned of security flaws in its e-business suite of applications that could allow hackers to create a buffer overflow and cripple the browser interface program. IBM had to fix security problems in its iNotes product and Domino servers, made by its Lotus division.

In the open source community, Apache, the leading Web server, had flaws related to a potential stack buffer overflow and remote access features. Sendmail, the commonly used Internet email server, didn't properly check long email addresses and this flaw could allow a hacker to gain control over the server. The Secure Sockets Layer (SSL) encryption software had an extensive list of problems that led to insecure implementations. Even the antivirus software company Symantec had a security flaw due to a feature that allowed users to access a free online service to check their computers.

In sum, software companies should be held responsible for security flaws and other defects, but within reason. Software products are complex to design and harder to test. No one has yet built a flawless software product of any size on the first try. And it will take more than a class-action lawsuit to fix human behavior when we live in an imperfect and open world. C

References

1. Cusumano, M. et al. Software development worldwide: The state of the practice. *IEEE Software*, (Nov.–Dec. 2003).
2. Keefe, T. Software insecurity. *Computerworld* (Aug. 5, 2002); www.computerworld.com.
3. Lohr, S. Product liability lawsuits are new threat to Microsoft. *The New York Times* (Oct. 6, 2003), p. C2.
4. Songini, M. J.D. Edwards user wins arbitration case against ERP vendor. *Computerworld*, (Feb. 21, 2002); www.computerworld.com.

Michael Cusumano (cusumano@mit.edu) is the Sloan Management Review Distinguished Professor at the MIT Sloan School of Management.