by Gary McGraw

How Flawed Is Microsoft?

Even though Microsoft has spent hundreds of millions of dollars on software security, company representatives still expressed great surprise when the Windows Metafile (WMF) vulnerability surfaced. There's a simple reason for this. Microsoft's approach, commendable in many ways, involves an overemphasis on code-level bugs and is thus subject to a major blind spot: overlooking architectural flaws such as the WMF problem.

Microsoft isn't alone. In fact, the WMF story provides a cautionary tale for anyone working on software security: Pay attention to software architecture, or suffer the consequences.

DEFECTS, BUGS, AND FLAWS—OH MY!

Software defects that lead to security problems come in two varieties. On the one hand, *bugs* are defects that can be found by looking at code. They usually involve misuse of APIs, buffer overflow problems, integer overflows, simple timing errors, and so on. You can find bugs using code review techniques (one of my software security touchpoints). One challenge is that there are literally hundreds of different kinds of bugs that can sneak into your implementation. Code-scanning tools such as Fortify Source Code Analysis are advanced enough to uncover potential bugs automatically so that they can be eradicated during development.

Flaws are best found by pondering design issues that go far beyond code.

On the other hand, *flaws* are software defects that exist at the architectural level. Flaws are much more difficult to spot and are best found by pondering design issues that go far beyond code. Fortunately, my architectural risk analysis touchpoint specifically targets flaws.

The WMF problem provides a prime example of a software security flaw. At its heart, the WMF problem is caused by a software feature being used in an unintended way. WMF files, designed in the late 1980s, allow image files to contain code that can be executed as the image decodes. Microsoft put this "feature" in on purpose. The problem is, nobody put on their black hat and thought through what an attacker might be able to do with such an inherently dangerous feature. Malicious hackers use WMF information to install rootkits, spyware, and other malicious code on their victims' machines. Some security experts estimate that at least a million computers have been compromised this way.

If you divide software security defects into bugs and flaws, you'll find the ratio to be about half-half. Mike Howard (Microsoft's outstanding software security guru) and I have discussed this division many times. Microsoft spends most of its energy looking for bugs, which is the low-hanging fruit. Its prefix tool, for example, is an advanced bug finder. Unfortunately, a myopic focus on bugs leaves flaws to fester. It doesn't help when Microsoft security people like Debby Fry Wilson say ridiculous things about how flaws such as WMF can't be anticipated. They can—they're just harder to find.

THE BUG PARADE

Too many software security experts are obsessed with bugs. That's because they're code junkies. By writing books that provide hundreds of code-based examples of bugs and no examples of flaws, these experts fall prey to the bug parade problem. Sure, we must understand and avoid buffer overflows, so talking about them is important. But at the same time, we must also think about flaws, including the misuse of "features."

No matter what software process you follow, you can adopt a simple but comprehensive set of software security touchpoints that adapt to, but don't radically change, the way you build software (for more details on this, see my book *Software Security* [Addison-Wesley, 2006]). Software security is more important than ever, but an exclusively bug-o-centric approach is fundamentally flawed.

Gary McGraw is CTO of Cigital, a software quality management consultancy. He is co-author of *Exploiting Software* (Addison-Wesley, 2004), *Building Secure Software* (Addison-Wesley, 2001), and *Java Security* (Wiley, 1996). His most recent book is *Software Security* (Addison-Wesley, 2006). Reach him at gem@cigital.com.