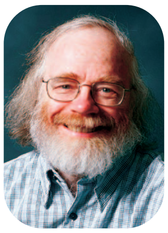
DRM, Complexity, and Correctness



STEVE BELLOVIN
Columbia University

Peter Gutmann recently posted an essay entitled "A Cost Analysis of Windows Vista Content Protection" on the Web (www.cs.auckland.ac.nz/~pgut001/pubs/vista_cost.txt). It's a fascinating summary of the implications of Vista's content protection mechanisms on system design. I'm not going to talk about whether Guttmann's analysis is correct in every detail or not (though I've known him several years and have a great deal of respect for him), and I'm certainly not going to wade into the digital rights management (DRM) wars now. Instead, I'm going to focus on complexity and security.

It's a truism that complex systems are less secure than simple ones. In some sense, this is inevitable because they contain more code that can be buggy. Worse, complexity (and hence potential incorrectness) increases more than linearly with the program's size. But it's not that simple—a certain amount of code is necessary for security. For example, an OS that didn't check passwords would have less code than one that did, but it would clearly be less secure. More realistically, code to perform array bounds checking is an important defense against buffer-overflow attacks, but it's also more code.

The real issue is with increased interaction among different pieces of code, which is the real danger with DRM. The DRM problem is very hard; attackers can try to recover content at many different points. Thus, as Guttmann's article points out, a video card has to know that it's displaying protected content and shut down outputs that talk to unprotected monitors. This communication path doesn't need to exist in the absence of DRM; the obvious question is whether it poses any new security problems, such as whether or not the new code has any buffer overflows. More subtly, we might wonder if an attack program can somehow assert to the system that it's processing protected content, thereby forcing the system to turn off the display. What if a Web page includes a URL that points to a protected but silent "song"? While this song is "playing," other functionality will be disabled. Some such mechanisms must exist if the ability to download protected content is to be preserved; the only question is how to guard against abuse.

Another threat comes from a set of measures designed to protect against weak—that is, buggy—device drivers. The ability will exist to "revoke" them. But again, how is this ability protected? The obvious solution is for the revocation message to be digitally signed; I think we can safely assume that this is the approach taken. However, is the code correct? Is the process that authorizes driver revocation correct? Remember the fraudulently obtained Microsoft code-signing certificates in 2001? Could something like that happen again?

I'm not, of course, asserting that any of these security holes actually exist. Microsoft has made great strides in securing its software, and I'm sure it has put a lot of effort into testing all the new DRM code. But as we all know, testing is dicey. Edsger Dijkstra once noted that "program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence." A lot of new mechanisms have been introduced; more seriously, a lot of new communications paths and dependencies have been introduced. Worst of all, these paths and mechanisms are solving a new problem, one with which the profession has very little experience. Did Microsoft get it right?

DRM may not be evil. It is, however, very, very complex, and, historically, complexity has led to insecurity. When contemplating such steps, we should bear in mind Dijkstra's dictum, one taken from his Turing Award lecture: "We shall do a much better programming job, provided that we approach the task with a full appreciation of its tremendous difficulty ... and approach the task as Very Humble Programmers." ☐

Steve Bellovin is a professor of computer science at Columbia University. He has a BA from Columbia University and an MS and PhD in computer science from the University of North Carolina at Chapel Hill. Bellovin helped create netnews, or usenet news, and is coauthor of Firewalls and Internet Security: Repelling the Wily Hacker *(Addison-Wesley, 2003). Contact him via www.cs.columbia.edu/~smb.*