Digital Rights Management, Spyware, and Security

In October 2005, Sysinternals' Mark Russinovich discovered a rootkit on his computer, which he later determined stemmed from a Sony-BMG compact disc. In this article, the authors examine the copy-protection software found on those discs and the implications for digital rights management.

Edward W. Felten and J. Alex Halderman
Center for Information Technology Policy, Princeton University

In late October 2005, Mark Russinovich of Sysinternals ran a routine security scan on one of his computers. Russinovich, an independent researcher and well-known expert on Windows internals, found what appeared to be a rootkit: malicious software designed to hide evidence of a system intrusion. Determined to understand how his system had been infected, he set to work diagnosing the infection. In an impassioned blog post (www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html), he described his mounting anger as he gradually discovered that the harmful software had been installed by a compact disc (CD) shipped by Sony-BMG, the world's second largest record company. For more details on Russinovich's discovery, see the "Revealing rootkits" sidebar.

As news of his discovery raced across the Internet, other researchers (including this article's authors) accelerated their own investigations that were already in progress. A series of revelations about Sony-BMG's discs followed. Two separate anti-copying technologies, shipping on separate Sony-BMG CDs, turned out to cause serious security and privacy problems for consumers.

Although this story is far from over, we can already see that it will have a far-reaching effect on the fate of anti-copying technologies, and on music and movie companies' market strategies in general. It's an instructive story, too, about the dark side of anti-copying technology.

Copy-protecting CDs

The CD format has survived for more than 20 years as a straightforward way of distributing music in digital form. To the chagrin of the music industry, consumers can easily use their computers to "rip" music files from CDs, encode the files compactly in the MP3 format, and then redistribute them over the Internet. For years, the music industry has sought technologies that could somehow hinder ripping or redistribution, and, in fact, some record companies sell discs that use such copy-protection technologies.

Sony-BMG has shipped discs with two such systems: XCP (short for extended copy protection) from the British company First4Internet, and MediaMax from the US company SunnComm. Both technologies target Microsoft Windows computers, but they pose no real barrier to copying on other operating systems, such as MacOS or Linux. MediaMax, for example, ships discs with a MacOS version of the copy-protection software, but it installs only if the user seeks it out. Because users have no reason to install the software, virtually no MacOS users will ever do so; thus, it won't affect the vast majority of MacOS systems. As of the end of 2005, XCP has shipped on approximately 20 Sony-BMG titles, and MediaMax on approximately 50 Sony-BMG titles, along with titles from smaller record companies. Each technology was shipped on millions of individual discs (exact numbers were not available at press time).

Copy-protecting CDs is inherently difficult. CDs store music files in compact disc digital audio (CDDA) format, which a wide range of devices can read. If the music is encrypted or stored in some other proprietary format, ordinary audio CD players won't be able to read it, and the disc will be useless to most customers. Backward compatibility requires that the music be stored in a format that existing computer software can read.

Revealing rootkits

Mark Russinovich uncovered the XCP rootkit by scanning his system with a rootkit detector program called Rootkit Revealer that he coauthored with Sysinternals partner Bryce Cogswell. Many rootkits hide files and programs by intercepting the system API calls that applications use to list running programs or see the contents of a directory. Before output from these API calls is passed on to other programs, code in the rootkit filters out the names of any files or processes that the rootkit author wants to hide, rendering them invisible to normal applications.

Rootkit Revealer is able to uncover such rootkits by turning their stealthiness against them. First, it lists all the files that are visible using regular system API calls. Then it examines the disk directly, bypassing the APIs that a rootkit would intercept. Discrepancies between these two accounts of the system raise red flags that a rootkit might be installed. Any files that are present on the disk but don't show up using the API calls are likely being concealed by a rootkit. Rootkit Revealer is available as a free download from www.sysinternals.com.

A CD's copy-protection technology must somehow stop software on a computer from reading the music files, even though the files are stored in a readable format. There are two types of methods for doing this. *Passive protection* tries to exploit subtle differences between how audio CD players read discs—and how computers read them—by putting things on the disc that attempt to confuse computers without affecting ordinary players. Space doesn't permit a full discussion of passive protection here, but it will suffice to say that passive protection systems aren't viable because computers aren't easily confused. To our knowledge, purely passive CD DRM technologies are no longer used on retail CDs, although some systems combine passive protection with active measures. *Active protection* accepts that the computer will read all music files on the disc, but it installs software on the computer that actively interferes with most attempts to read the disc. For example, a device driver might be replaced by a modified driver that intentionally garbles any music files read from the disc.

Both XCP and MediaMax use active protection and ship with a special music player application that bypasses it and accesses music files without interference; however, these special music players aren't programmed to do anything but play music or translate it into certain encrypted formats. This not only prevents users from illegal copying, but also from many lawful uses of the music, such as downloading it into an iPod.

In designing an active protection system, a key technical problem is how to get the protection software installed on the user's computer. Users aren't accustomed to installing software just to listen to a CD, and if the software isn't installed, it can't do its job. Both XCP and MediaMax rely on a Windows feature called "autorun," which is enabled on recent Windows versions by default. (Other systems, including MacOS and Linux, have no autorun feature, which is one reason why XCP and MediaMax don't affect them.) Autorun lets CD-ROMs display splash screens when they're inserted. It also lets copy-protection software run automatically. If a CD is inserted into the computer, autorun (assuming it's enabled) will look on the CD for a file with a certain name, and if it finds that file, it loads into the computer's memory and executes as a program. Autorun is dangerous because it can let programs execute without the user's knowledge or consent. It's a good security practice to disable autorun; for instructions, see www.annoyances.org/exec/show/article03-018.

In adopting active protection, Sony-BMG crossed an important boundary: it began distributing software, thus its CDs could inflict serious security and privacy harm on its customers. An unprotected or passive music CD can't expose customers to security exploits, install spyware, or leak customer information, but executable software can do all of these things, unless it's designed and tested with security in mind—as Sony quickly discovered.

The XCP rootkit

When XCP installs its active anti-copying program, it also installs a second component that hides the software's existence. Normally, programs and data aren't supposed to be invisible, particularly to system administrators; they can be superficially hidden to simplify a novice user's view of the computer, but administrators must be able to see what's installed and running to secure the computer. What kind of software would want to hide from system administrators? Typically, viruses, spyware, and rootkits—malicious programs that cover a remote intruder's tracks—fit this bill. Rootkits in particular are known for their stealthiness, and they sometimes go to great lengths to conceal their presence.

Rootkits on Windows often hide files and programs by modifying the jump tables that Windows uses to find the code for various system functions. An entry in the jump table corresponds to a particular function that Windows provides. By modifying the jump table, a rootkit can cause invocations of that function to be handled by the rootkit's own code, rather than the normal Windows code. This allows the rootkit to change the behavior of basic Windows functions, such as the ones that list running applications or a directory's contents.

The XCP rootkit modified the system so that any file,

Further reading

We've listed a few links where you can find more information on the Sony situation and rootkits.

- Electronic Frontier Foundation's Sony BMG Settlement FAQ at www.eff.org/IP/DRM/Sony-BMG/settlement_faq.php.
- The authors' Freedom to Tinker weblog at www.freedom-to-tinker.com. For specific information on CD DRM, see www.freedom-to-tinker.com/?cat=30.
- G. Hoglund and J. Butler's book *Rootkits: Subverting the Windows Kernel*, Addison-Wesley, 2005.
- S. Ring and E. Cole's "Taking a Lesson from Stealthy Rootkits," *IEEE Security & Privacy*, vol. 2, no. 4, 2004, pp. 38–45.
- Mark Russinovich's Sysinternals Blog at www.sysinternals.com/Blog/. See especially www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html and following entries.

Windows registry entry, or running program whose name started with the string "sys" became invisible. XCP itself uses names beginning with that prefix, so this trick's apparent intent was to make XCP components harder for the user to find and remove.

But the rootkit had at least two harmful side effects. First, its design, based on well-known malware methods, tends to trigger security alarms, as it did on Russinovich's system. Even if the rootkit were harmless, these alarms would be a problem. Second, the rootkit created a hiding place for other malware. Any hostile program, whether privileged or not, could hide from the user (and the user's security tools) by giving itself a name that started with the magic string "sys." Attackers would no longer need to create their own rootkits and find a way to install them—Sony-BMG already did the work for them.

Even beyond the rootkit's direct impact, its deployment on CDs—which don't need to contain any software at all—struck many users as an underhanded tactic, below the standard expected of a large company like Sony-BMG. Users were angry, but things would soon get even worse.

XCP and MediaMax as spyware

In the first half of November 2005, researchers (including Russinovich and the authors of this article) turned up more information about XCP and MediaMax, and it was soon clear that both programs were spyware.

Although the precise definition of "spyware" is often debated, what's clear is that the term applies to software that's installed without the user's informed consent, is difficult or impossible to uninstall, and transmits information about the user's activities without notice or consent. Both XCP and MediaMax meet this definition.

Installing without consent

When an XCP disc is inserted into a Windows machine, the disc uses autorun to load into memory the active protection software that tries to interfere with attempts to read the disc. It also displays an end user license agreement (EULA), which the user is asked to accept. If the user agrees to the EULA, the full XCP software is installed. If not, the disc is ejected and the program stops running.

The system exceeds user consent in at least two ways. First, the program is loaded and run before the user has agreed to anything. The user hasn't consented to the installation; most users certainly wouldn't expect such a thing to happen when inserting a music CD. Second, the EULA asks the user to consent only to installing "a small proprietary software program … intended to protect the audio files embodied on the CD," a description that can hardly be said to cover the full XCP system, including the rootkit and the "phone home" feature (described later). Furthermore, although the EULA disclosure refers to the audio files on the current CD, the installed software restricts access to audio files on all titles protected with the XCP system without asking for further consent.

Although the MediaMax system lacks a rootkit, its behavior exceeds user consent even more flagrantly. Like XCP, it loads and runs a program immediately when a disc is inserted, before the user even sees the EULA. But unlike XCP, MediaMax installs software permanently even if the user rejects the EULA—that is, it installs the software even when the user explicitly instructs it not to do so.

Resisting removal

Clearly, both XCP and MediaMax were designed to resist detection and removal—both ship without any kind of uninstaller. Initially, both manufacturers claimed that users who complained could get an uninstaller, but the process for doing this erected so many barriers as to be nearly impractical. (XCP users, for example, had to fill out a Web form containing personal information, then wait days for a reply email, then fill out another Web form and install more software, and finally wait days for yet another email, before receiving a URL from which to download an uninstaller that worked only for a limited time and only on the machine from which the second form was submitted.)

Both systems also took steps to resist detection of the software. XCP uses the rootkit method described earlier, and MediaMax makes its device driver invisible from the normal user interface for controlling Windows drivers.

Transmitting information about user activities

The last piece of our spyware definition is that the software "phones home"—that is, it sends information about the user's activities back to the vendor. This, too, was true for both XCP and MediaMax systems.

Both systems were designed to contact a vendor Web site whenever the user inserted a protected disc. The ostensible purpose of this was to download images or advertisements that would be displayed while the music played, but it also created entries in the vendor's Web server log, noting the users' IP addresses, disc inserted, and the times and dates it was inserted. Despite this rather obvious breach in privacy, the vendors' Web sites claimed that they didn't gather information about users' activities.

From bad to worse

These spyware revelations added to the public outcry. Sony, on the defensive, reversed course, recalling the XCP discs and offering to replace customers' discs with ordinary, unprotected CDs. The company also decided to release unrestricted uninstallers for XCP and MediaMax software, so that users who objected to the programs could remove them. Although intended as a sign of good faith, this decision only fanned the flames further—both uninstallers ultimately opened serious security holes on users' machines.

Both vendors chose to deliver their uninstallers via scriptable ActiveX controls—downloadable programs suitable for embedding in a Web page. In both cases, the enclosing Web page would invoke the uninstaller control, passing it a URL from which to download the uninstaller code; the control would then download code from that URL and execute it.

Incredibly, neither ActiveX control checked whether it was passed an approved URL or whether the downloaded code was approved. Instead, the controls were programmed to download and run code from any URL they received. The result was that any malicious Web page could include a vendor's ActiveX control and then instruct the control to download and run code from a malicious site. The controls designed to deliver the uninstaller could just as well deliver attack code.

To make things worse, the uninstallers didn't remove the buggy ActiveX controls, but left them in place—meaning the user's vulnerability would persist long after the XCP or MediaMax software had been removed. For one vendor to make these sorts of simple errors was embarrassing, but for both vendors to make them simultaneously was an amazing, and unfortunate, coincidence.

Enter the lawyers

What started as a technical issue for Sony became a customer-relations one—and it was about to become a legal issue. Several class-action lawsuits were filed on behalf of consumers, claiming harm due to the initial spyware and the subsequent security problems. Governments began investigating, too: the Texas attorney general filed a suit under the state's anti-spyware law, and authorities in Italy, Florida, and New York, among others, were reportedly considering action.

In late December 2005, lawyers for one of the class action suits announced a preliminary settlement in the primary class-action suits. Under the settlement, Sony would offer compensation to those who bought the affected discs and would stop shipping discs that contained the affected technologies. This didn't resolve all the legal issues—a court must still approve the settlement, and any government actions, including the Texas suit and an investigation by the Florida Attorney General, are ongoing.

The settlement is an important step in Sony-BMG's effort to climb out of the hole it dug, but what lessons can we learn from this incident?

What went wrong?

The first question we must ask is why things went wrong. Was this just an anomaly—a speed bump on the road to an effective, unobtrusive DRM future—or was it a sign of deeper problems with DRM?

It doesn't look like an anomaly. For starters, these problems didn't affect just a single DRM system—rather, they applied to two separate systems (XCP and MediaMax), developed by rival companies, both of which turned out to contain dangerous spyware, in strikingly similar ways. Is this a coincidence?

We think it isn't. By looking carefully at CD copy protection as a technical problem, we can see why DRM designers are drawn to spyware tactics as their best hope of halting copying. As we mentioned earlier, CDs store music files in CDDA format, which a wide range of devices can easily read. If the music is encrypted or stored in some other format, ordinary audio CD players won't be able to read it, and the disc will be useless to most customers.

By looking carefully at CD copy protection as a technical problem, we can see why DRM designers are drawn to spyware tactics.

Experience teaches that purely passive protection schemes, which try only to change how information is formatted on a disc, won't be able to stop computers from reading discs. Any effective scheme thus must use active protection—it will have to install software on the user's computer, and that software must actively interfere with attempts to read the disc, such as by corrupting the data stream coming from the disc.

Even if all of the affected discs are recalled, many end users will be left with the dangerous software still installed on their systems.

Let's look at an example. Suppose a user wants to use the iTunes application to read a disc, but the DRM vendor wants to prevent the user from doing so because iTunes can be used to copy the disc. The active protection software detects this and interferes to ensure that iTunes gets a garbled copy of the music. (Of course, iTunes has its own copy protection scheme, which is far from perfect.)

The key issue is that active protection only works if the DRM software is running on users' computers, but because it holds no value to them, they don't necessarily need to install it. It only stops users from doing things they want to do (such as listening to music with iTunes) and exposes them to attacks if the software is buggy.

When designing a CD DRM system based on active protection, you face two main technical problems:

- You must get your software installed, even though the user doesn't want it.
- Once your software is installed, you must keep it from being uninstalled, even though the user wants it gone.

Crucially, these are the same two technical problems that spyware (or other malware) designers face. And people who face the same technical problems tend to find the same technical solutions. How do you get software installed against the user's wishes? You mislead the user about what is being installed and the consequences of installation, or you execute installation without obtaining permission. How do you prevent software from being uninstalled? You don't provide an uninstaller, you provide one that doesn't uninstall the whole program, or you attempt to cloak the software so the user doesn't even know it's there.

Having set off down the road of CD copy protection, Sony-BMG shouldn't have been surprised to have arrived at spyware. That's clearly where the road leads.

Remaining holes

What's most surprising in this case is that Sony-BMG chose to deploy these technologies despite their many obvious weaknesses. As we described earlier, the systems rely on active protection, which means installing software on the user's computer. The software must be installed immediately on insertion of a protected disc—otherwise the user will be able to rip an unprotected copy of the music before the software is ever installed. On Windows, the only way to do this is via the Windows autorun feature, which allows code from the disc to be run as soon as the disc is inserted.

Unfortunately for DRM vendors, autorun is easily disabled. Windows has an option to turn off autorun permanently, and many people do turn it off to prevent inadvertently installing code when a CD-ROM (or other disc) is inserted. Even if autorun is enabled, it can be shut off temporarily: holding down the Shift key while inserting a disc will prevent the disc from autorunning. The result is that both XCP and MediaMax can be defeated by simply holding down this key.

Another well-known trick for defeating CD copy-protection technologies is to block the reading of the outside edge of the disc's data area by covering it with opaque tape or drawing over it with a felt-tipped pen. This effectively transforms a protected disc into an unprotected one because the unprotected music files are stored toward the center of the disc and the copy-protection software is stored toward the outside edge.

The last failure mode for these systems is the simplest of all: they only work on Windows systems. Because MacOS and Linux lack the autorun feature, active protection software can't install itself on these systems, unless the user does so voluntarily.

Business lessons

From a nontechnical viewpoint, Sony-BMG's experience has much to teach the music industry. The most important lesson is that DRM can have serious side effects, especially relating to security and privacy.

Salesmen often present DRM as a technology that prevents piracy and does nothing else. Although few music executives today accept this simple-minded view, many do seem to see DRM as having only minor side effects that consumers will learn to accept. Sony-BMG's technologies are counterexamples that challenge this view. Thus far, two of the three major CD DRM technologies have turned out to cause serious side effects.

It's worth noting, too, that the problems caused by these misbehaving DRM technologies will persist. Even if all of the affected discs are recalled, many end users will be left with the dangerous software still installed on their systems. Some won't realize that they inadvertently installed software just by listening to a music CD. Many corporate computers will have had the software installed by employees who brought CDs to work. Cleanup is left to users, and corporate IT departments, who have to detect the software, download cleanup tools, and use the tools correctly. If experience with other dangerous software is any guide, many users will remain vulnerable until they retire their computers or reformat their hard drives.

The second business lesson is that the public, the press, and the legal system will hold record companies responsible for overreaching. Sony-BMG tried initially to shrug off the problems, but this strategy failed and the storm mounted until the company had to retreat at a significant cost. Legal expenses and the product recall must have cost tens of millions of dollars. The price of lost customer trust is even harder to reckon.

The clearest lesson of all is that record companies must be much more careful about which DRM technologies they adopt. As Sony learned, the stakes are much too high to leave the issue to small technology vendors. If record companies are going to ship software, they better learn to think and act like software companies.

Sony-BMG and other record companies face some hard decisions. The Internet is transforming the distribution and promotion of music, and the industry is searching intently for new business models.

Many in the industry see DRM as part of the industry's future. For these DRM advocates, Sony-BMG's misadventures with CD copy protection are a cautionary example of the downside of DRM. Sony-BMG lost both money and customer goodwill, without preventing piracy. Will future DRM systems do better? We think not; but some in the industry disagree.

The only thing we can predict with confidence is that the industry's DRM strategy will be more cautious and incremental. With so much at stake, that's a sensible strategy. □

Edward W. Felten *is a professor of computer science and public affairs at Princeton University and director of Princeton's Center for Information Technology Policy. His research interests include computer security and privacy, and information technology policy. Felton has a PhD in computer science and engineering from the University of Washington. Contact him at felten@cs.princeton.edu.*

J. Alex Halderman *is a PhD student in the Department of Computer Science at Princeton University. His research interests include computer security, digital rights management, information privacy, and the interplay between technology and public policy. Halderman has a undergraduate and masters degrees in computer science from Princeton. Contact him at jhalderm@cs.princeton.edu.*