BEFORE THE COPYRIGHT OFFICE OF THE LIBRARY OF CONGRESS

Docket No. RM 2008-8: In the Matter of Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies

Comment in Support of Proposed Exemptions 8A and 8B

Research Plan:
Side-Effects of the SecuROM DRM System

J. Alex Halderman
Assistant Professor of Electrical Engineering
and Computer Science, *University of Michigan*

CSE Building, Room 4717
2260 Hayward Avenue
Ann Arbor, MI 48109-2121

Summary of Argument

In support of my previously submitted requests for exemption from the anti-circumvention measures of the DMCA numbered 8A ("*Literary works, sound recordings, and audiovisual works accessible on personal computers and protected by technological protection measures that control access to lawfully obtained works and create or exploit security flaws or vulnerabilities that compromise the security of personal computers, when circumvention is accomplished solely for the purpose of good faith testing, investigating, or correcting such security flaws or vulnerabilities*") and 8B ("*Video games accessible on personal computers and protected by technological protection measures that control access to lawfully obtained works and create or exploit security flaws or vulnerabilities that compromise the security of personal computers, when circumvention is accomplished solely for the purpose of good faith testing, investigating, or correcting such security flaws or vulnerabilities*"), I submit this document, a research plan detailing the real-world steps I would like to pursue with my students in studying the controversial SecuROM DRM system. The execution of the plan would likely add to the mounting evidence that PC-based DRM systems are inherently likely to carry security vulnerabilities, but the plan would be chilled in many respects by the DMCA, thus denying essential information and protection to the PC-using public.

Introduction

Security research studies how computers are attacked in order to devise new ways to defend them. As I tell the students in my security class, the only way we can hope to build systems that won't fail under attack is by understanding how our adversaries think and learning how they operate. In other words, know your enemy — it's as important for us as it was for Sun Tzu.

Within this framework, attacks facilitated by DRM systems are of considerable scientific interest. Software security flaws are not unusual, but mounting evidence suggests that DRM systems are especially likely to be vulnerable and that their vulnerabilities are abnormally likely to be severe. In studies of the XCP and MediaMax audio CD copy protection systems, I showed that side effects of those DRM systems caused threats to users' security and privacy.[1] More recently, attackers discovered how to exploit problems in SafeDisc, a DRM system for PC games, to seize control of computers.[2] In light of these threats, characterizing the risks of DRM systems and understanding the causes of those risks are increasingly high priorities for security research.

Many scientists theorize that elevated security threats are an inherent risk of DRM systems. One reason for this is that the goal of DRM systems — preventing users from accomplishing certain tasks on their PCs — is in conflict with primary security principles, which require PC owners to be informed and able to control the operation of their PCs. Another is that DRM systems must often employ risk-prone software techniques, such as overriding the restrictions that operating systems place on typical software. This theory predicts that future DRM systems will cause similarly severe security and privacy problems. We can test this prediction, and find evidence to support or refute the theory, by studying the security of other DRM systems.

I am considering launching such a study in collaboration with some of my students. One appealing target is a PC-based video game DRM system called SecuROM.[3] Our goals would be 1) to identify any security or privacy problems resulting from use of the system, and 2) to better understand the risks associated with DRM systems in general. This document describes the approach we would follow in such an investigation. It also gives several examples of how potential legal risks due to the DMCA could complicate the process.

Approach

An academic study of SecuROM or a similar PC game DRM system would likely proceed in the following stages:

Stage 1: Preparation – *Studying the Context*

We would begin by conducting background research to determine what is already known about the DRM system. This preliminary inquiry would examine prior security studies (if any), users' reports about negative effects, and statements from the manufacturer. We would evaluate the credibility of each piece of information and pose a list of unanswered questions to address in our investigation. Based on this research, we would select PC game titles that use the DRM system and purchase them for study.

---

[1] J. Alex Halderman and Edward W. Felten, *Lessons from the Sony CD DRM Episode* (Feb. 14, 2006), *available at* http://www.cse.umich.edu/~jhalderm/pub/papers/rootkit-sec06-full.pdf.

[2] Microsoft, Security Bulletin MS07-067 – Important: Vulnerability in Macrovision Driver Could Allow Local Elevation of Privilege (Dec. 11, 2007), available at http://www.microsoft.com/technet/security/Bulletin/MS07-067.mspx.

[3] Investigating a second game DRM system would allow us to compare any problems we found to the attacks discovered against SafeDisc. Security investigations often adopt such a comparative approach, since it helps distinguish implementation-specific mistakes from inherent risks.

Stage 2: Instrumentation – *Lifting the Curtain*

Once we acquired the games, we would investigate techniques for observing the behavior of the DRM system and its interaction with the PC. These would involve running the games on computers that have been instrumented with software or hardware for monitoring their operation. We would likely use off-the-shelf system monitoring tools (like Microsoft's *Filemon*, *Procmon*, and *Regmon* applications), software engineering tools such as debuggers (which allow execution tracing, condition testing, breakpoint execution, and dynamic review and modification of main memory), and standard computer forensics techniques like hard disk imaging, virtualization, and live memory analysis.[4] Since DRM systems are frequently designed to resist attempts to monitor and understand their behavior, we would likely encounter circumstances that would require us to develop additional new techniques.

Even this early phase could potentially involve legal risks. In order to conduct controlled experiments, we would need to develop ways to reproducibly invoke different DRM behaviors. However, SecuROM and many similar DRM systems attempt to limit how many times a game can be installed or activated, sometimes permitting as few as three activations. Our investigation would be severely hampered if we could only observe these operations a small number of times. The most straightforward technical solution would be to defeat this part of the DRM so that the system would behave as if it had not reached the usage limit, but it is unclear as to whether this would violate the DMCA anti-circumvention provisions. Such a concern would threaten to halt our experiments and force us to seek the advice of counsel.

Stage 3: Experimentation – *Understanding the Design*

Next, we would use these monitoring techniques to conduct tests under a variety of usage conditions and observe the DRM system's behavior. We would develop hypotheses about how the system works — what files are installed, what their purposes are, how the system stores data, how the system interacts with hardware, etc. — and conduct experiments to confirm our hypotheses. At the end of this stage, we would aim to have a detailed understanding of the system's design and operation.

As in the previous stage, we would expect to face questions about potential liability under the DMCA. Much of the functionality of a DRM system only comes into play when someone attempts to circumvent it. In order to understand these aspects of the system, it would likely be necessary to circumvent parts of the DRM in our tests. Accordingly, we would likely need to involve our legal team in the design of these experiments.

Stage 4: Assessment – *Identifying High-Risk Targets*

After we understood the operation of the DRM system, we would assess which parts of the system would be most likely to contain harmful security flaws. For flaws to be harmful they must be accessible to an attacker; thus, we would analyze the system's "attack surface" – that is, the software components

[4] These are same kinds of tools that we used to study the XCP and MediaMax audio CD DRM systems. Researchers have applied them to many kinds of security investigations, including California's groundbreaking "Top-to-Bottom" electronic voting systems review. *See State of California Standard Agreement #06158101* p.5-6, *available at* http://www.sos.ca.gov/elections/voting_systems/ttbr/sos_uc_contract.pdf.

and interfaces that would be exposed to attack under different threat scenarios. Design mistakes and programming errors in these components would likely be dangerous.

We would also assess the level of security risk associated with different design choices. Experience with security failures shows that certain designs are especially risky or brittle. For example, consider "buffer overflow" vulnerabilities, a common security flaw that allows an attacker to insert malicious code into a running program. In normal applications these flaws often cause only minor harm, because the computer's operating system limits the hardware resources and data files that the malicious code can access. Yet many DRM systems install software that integrates deeply into the operating system kernel, bypassing these limitations. As a result, a similar flaw in such a system could give attackers complete control of the computer. By identifying elements of the DRM system design carrying elevated security risks and potentially exposed to attackers, we could focus the remainder of our investigation on the areas that most likely to be vulnerable.

Stage 5: Probing – *Searching for Vulnerabilities*

Next, we would search for vulnerabilities in the high-risk components. There would be several possible approaches. One would be to manually inspect the components for programming mistakes and for design patterns that have been exploitable in other contexts. Such problems are often subtle, so this kind of investigation would rely on the skill and creativity of the investigator. Another approach would be to build automated testing tools that could repeatedly run the software with inputs chosen at random according to carefully designed rules. This technique, known as "fuzzing," could detect potentially exploitable bugs by bringing about conditions that the programmers of the DRM system did not anticipate.

Certain kinds of vulnerabilities might only be exposed after parts of the DRM system had been circumvented. This need to circumvent does little to reduce the severity of these vulnerabilities — attackers are generally not law-abiding, so we can assume that they would not be deterred by the DMCA from defeating the DRM system in order to conduct an attack. However, the need to circumvent the DRM system would create potential risks for my students and me as we searched for these vulnerabilities. The possibility of liability under the DMCA would likely add considerable complication and delay to this stage of the investigation, and it might force us to leave possible problem areas uninvestigated.

Stage 6: Confirmation – *Demonstrating Attacks*

We would then experimentally confirm whether an attacker would be able to exploit each vulnerability we identified. The standard approach in the security field is to attempt benign demonstration "attacks" using the vulnerabilities. If we could successfully attack test computers in our lab, it would establish that real attackers could likely cause significant harm.

Certain attacks might only be possible after circumventing parts of the DRM. In order to test these we would probably need to consult with our lawyers and again weigh the risk of potential liability under the DMCA.

Stage 7: Evaluation – *Considering Implications, Solutions, and Lessons*

The final stage would involve understanding the consequences of any attacks we found. Important questions would include the magnitude, scope, and potential targets of the attacks. We would consider what the manufacturer would need to do to fix the problems, and what users could do to defend themselves while they await the fix.

In some past instances where DRM systems caused security problems, such as the XCP rootkit, the simplest way for users to defend themselves involved disabling or circumventing parts of the DRM. If we were to face a comparable situation with SecuROM or a similar DRM system, we would need to circumvent the DRM ourselves in order to develop and test these defenses. Unfortunately, the potential for DMCA liability could force us to involve our attorneys once again, and might force us to avoid developing such defenses.

Potential Impact

Finally, we would draw broader lessons from our findings and consider questions such as the following: What were the underlying causes of the problems we found? How can future DRM implementations avoid making similar mistakes? Are there ways to build DRM systems that are less risky? Can we combine technological and non-technological tools to reduce the potential for DRM-related security harms?

Evidence of significant vulnerabilities would be a valuable addition to the growing research literature about collateral damage from DRM systems,[5] but scientists are not the only interested parties. Gamers, game developers, and DRM system vendors all have direct stakes in the security of these systems. Other PC users and policymakers should also take interest, since the collateral damage can extend beyond the PC where the DRM system is installed. Past studies have revealed security flaws in DRM systems that could be used to take control of the computer and use it for a variety of crimes, such as sending spam, attacking web sites, and distributing illicit content. Security flaws in PC-based video game DRM systems could be even more destructive. Since popular games are installed on tens of millions of PCs, an attacker who could take over a large fraction of them and could build an enormous "botnet," a virtual army under the attacker's control with enough network bandwidth to disrupt large portions of the Internet.

A rigorous study of side effects of SecuROM and similar DRM systems could reduce risks for millions of users and advance the state of the art in the security field. By discovering potential attacks, we could help DRM system developers make repairs before users are harmed and help users understand and correct the risks themselves. Our findings might also reveal new classes of threats and point to other DRM systems that should be checked for similar problems. By comparing flaws in SecuROM and similar DRM systems to other instances of security harm from DRM systems, we might be able to gain insight

[5] A paper describing the problems and their implications would be appropriate for publication in a leading academic security forum, such as *ACM Computer and Communications Security*, *IEEE Security and Privacy*, or *USENIX Security*.

into the causes of these problems and suggest changes that would help make future systems less dangerous.

Whether to Go Forward

Though this project falls squarely within my expertise and interests and could have a large positive impact, the exceptional uncertainties created by the DMCA undermine the project's potential. I fear that unforeseen legal risks might arise in a later stage of the project and might force my students and me to abandon our entire effort after months of work. Even when the legal questions are foreseeable, they are likely to create delays, increase costs, and multiply the complexity of the project. In some of my past studies, legal issues have forced me to trade proven experimental methods for ones that carry fewer legal risks but are more difficult or less effective, and a significant fraction of time and energy has been expended talking with lawyers instead of doing research.

The worst-case outcome, losing a major lawsuit, would be financially and professionally devastating. A few weeks ago, on a cold Michigan night, I lay awake past midnight worrying about the personal risks the project would entail, not only for me, but also for the students who trust my advice. Similar research has already left me starring down the barrel of a lawsuit. Yet, among those who are cognizant of the legal risks, I must be one of the most difficult to chill. Most researchers are not willing to put their families' homes on the line. Most researchers are not backed by teams of smart lawyers. These should not be prerequisites for conducting broadly beneficial science.

Accordingly, I reiterate my request for the aforementioned exemptions to the DMCA anti-circumvention measures.

Sincerely,

/s/

J. Alex Halderman