Some comments on reverse engineering and the Digital Millennium Copyright Act

Yaron Minsky

January 21, 2000

From my position as a graduate student in Computer Science and a user of both commercial and non-commercial software products, I am very concerned about the potential weakening of the right to reverse-engineer software products. The recent cases in New York and California involving DeCSS are what brought this matter to my attention.

I think it's important that Copyright Office understand the importance of reverse-engineering to the software industry in general, and in particular to the growing sector of Open-Source software. Indeed, I believe that reverse-engineering for the purposes of compatibility is critical for the health of the software industry.

A commonly commented upon feature of the software industry is the commonality of the *network effect*. The network effect occurs when a given product or standard becomes more valuable the more widely it is used. The classic example of this is the fax machine. The more people have a fax machine that follows a certain communication standard, the more valuable your personal fax machine that follows that standard becomes. In such situations, there tends to be a single standard that ends up as the winner. The market has to pick a winner, but it doesn't always pick the best one — sometimes being first, or being cheap, or being easy to get, is good enough.

The software industry's understanding of this fact can be seen from their actions. Many software companies try to get their software out early and cheap, even if it's buggy and lacks functionality. The goal is to get the "first-mover" advantage. When a company gains ubiquity, it gains an enormous amount of power in the market. Examples include standards like Windows, the TCP/IP networking protocols (used for routing virtually all internet traffic), Microsoft Word, and others.

When such a standard becomes dominant, it is extremely hard to dislodge. And often the only way to dislodge the standard product is to first

be compatible with it. Thus, any word processor that is to compete with Microsoft Word must be able to read and create Microsoft Word documents.

But that's where the problem comes in. Microsoft guards the Microsoft Word format vigilantly as a valuable trade secret. And as a result, programs like WordPerfect can only have partial compatibility with Word. The inability to be compatible leads to a virtual inability to compete. Thus, we are left with the continued preeminence of inferior software products.

Reverse engineering is a partial solution to these problems. Were it not for reverse engineering, most word processors would not be able to compete with Word at all. And if the Digital Millennium Copyright Act (DMCA) and its ban on reverse engineering is interpreted broadly, then competition in software will be stifled.

Reverse engineering is particularly important to Open Source software. Open Source software is software that is licensed on terms that allow anyone to use and modify the code for their own purposes. A number of multi-billion dollar companies, including SGI, RedHat, VA Linux, IBM and soon LinuxCare have businesses based on Open Source software. One of the key advantages of Open Source is that standards embedded in Open Source software are not the proprietary secrets of any one company, and so it makes an attractive standard to agree upon.

But despite the power of Open Source in establishing standards, closed standards sometimes win. And in that case, developers of Open Source software need to use reverse-engineering to compete. A few examples of Open Source projects that need to do reverse engineering:

- The *AbiWord* (http://www.abisource.com) word processing program. Reverse-engineering is used to figure out the Microsoft Word file format.
- The *Samba* file server (http://www.samba.org) allows for sharing of files between Unix servers and Windows clients. Samba is a big reverse-engineering project of Microsoft's SMB file server protocol.
- The *gPhoto* project (http://www.gphoto.org) allows for the use of digital camera with the Linux operating system. They use reverse-engineering to figure out how the camera work.

Despite the large amount of money in Open Source businesses, much, if not most, Open Source software is developed by individual volunteers. These individuals generally don't have access to lawyers or money enough to engage in litigation on these issues. As such, I think the Copyright Office

needs to make a clear statement that reverse engineering for the purposes of compatibility is virtually always legal under the DMCA, to make it clear that trying to sue Open Source developers out of existence simply isn't going to work. Lesser assurances will result in a chilling effect, with software authors fearful of jail time or the high fines allowed under the DMCA.

Thank you for reading this, and I hope my thoughts will help you come to your decision.