

## The Unreasonable Fear of Infection

by Lawrence Rosen

Whether you are an open source software provider, a corporate IT manager making hard decisions about new Linux deployments, or an attorney advising your client whether to adopt free and open source software, it is important to understand the implications of GPL licensing. In fact, no evangelizing about open source can be effective if you can't allay fears and misunderstandings about the GPL.

Just the other day, a lawyer friend of mine told me that her corporate clients are afraid of using open source software that is licensed under the General Public License (GPL). Those companies have been warned that the mere interaction of their own proprietary software with such software as Linux will require them to publish their source code. Those companies are considering whether someday to make their software open source, but they don't want to be forced into that step simply because they invite other open source software into their hallowed halls.

This fear of software infection reminds me of the early days of the AIDS epidemic. "Be careful of kissing or hugging," it was said, "because you can get the disease even from casual contact." It took years to calm people down. For people irrationally afraid of the GPL, I also urge calm. You can't catch the GPL simply by touching software.

The GPL is not a disease. It is a software license that is used successfully by over 70% of open source projects. It is designed to satisfy certain philosophical and economic objectives that are widely shared by many members of the open source community.

The specific language in the GPL that frightens some people is the following:

“2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

....

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this [GPL] License.

The phrase “work based on the Program” in section 2 of the GPL is further defined in the GPL. Such a work is “either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.”

Section 2(b) of the GPL, in light of the definition of “work based on the Program”, is sometimes described as “viral” or “infectious” because it affects any software, even the licensee’s own software, that “contains or is derived from” the original GPL-licensed program. A licensee’s own software, if it is a “work based on

the Program,” becomes subject to the terms of the GPL, thereby requiring the licensee to publish his or her own source code.

Set aside for the moment the philosophical argument that it might be good if all software source code were published. For many vendors of proprietary software that prospect is frightening. They do not want to be forced to publish their source code simply because they have used it in combination with GPL software.

The risk, however, is vastly overstated. The GPL doesn't use the word “combination.” It uses the phrase “contains or is derived from.” These terms have very specific meanings under copyright law that effectively reduce the reach of the GPL to a licensee's own code.

A “derivative work” is defined in the Copyright Act, 17 USC 101, as a:

“work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a ‘derivative work’.”

Simply combining a copyrighted work with another work does not create a derivative work. The original copyrighted work must be modified in some way. The resulting derivative work must itself “represent an original work of authorship.” So if the licensee doesn't modify the original GPL-licensed program, but merely runs it, he is not creating a derivative work.

Consider the scenario where the Linux operating system, a GPL-licensed program, loads and executes a proprietary program. The Linux program is not modified; it is merely used for the purpose for which it was designed. The proprietary program does not “contain” nor is it “derived from” Linux. Linux does not infect the proprietary program, and the proprietary program does not become subject to the GPL.

Software experts distinguish among various forms of program linking to create a combined program. Static linking requires a modification to the code of one program to allow it to link to another program. Such a modification, since it requires changes to the source code of the linking program, arguably creates a derivative work. If the linking program is licensed under the GPL, then the derivative work also becomes subject to the GPL.

Dynamic linking, on the other hand, is a transitory relationship between two programs for which they are each pre-designed. The linking program need not be modified to implement the linkage. For example, a printer driver for a new printer can be installed in a program without modifying the source code of the original program. Such linkage does not constitute the creation of a derivative work.

An even more tenuous relationship is established between programs that interact through data using a published application program interface (API).

Simply passing data between two programs, even if that data influences the behavior of those programs, does not create a derivative work of either program.

One important word of caution. There are subtle differences among the courts about the meaning of the term "derivative work." If you are a software developer, you should review the specific characteristics of your software with a knowledgeable attorney to see if, based on the case law in your jurisdiction, your "combined program" could be deemed a derivative work of a GPL program. In many instances, I'm confident you'll discover that the fear of being forced to publish your own code is exaggerated.

The GPL is an attempt to enforce an economic bargain between licensors and licensees. Licensors under the GPL open their source code and distribute their software freely to all who agree to do the same for their own derivative works. If a licensee creates a derivative work by modifying the original GPL-licensed program, or embeds the GPL-licensed program within his or her own program, the resulting work must also be licensed under the GPL. If there are no modifications to the GPL-licensed program, and it is not embedded within a proprietary program, the terms of the GPL simply don't apply to the licensee's software.

A derivative work is not created by merely touching, any more than one catches AIDS by merely hugging. A more intimate relationship is required. In copyright law terms, one must create "an original work of authorship." An author must consciously recast, transform, or adapt the GPL-licensed software (all of which are forms of "modification") before the GPL applies to the new software.

Even then, the proper term isn't "infection." The application of the GPL to derivative works is the enforcement of the terms of a bargain. You can create derivative works of GPL-licensed programs only if you agree to return the favor. For that reason, I prefer the more positive term "inheritance." A derivative work inherits the benefits of the GPL.

Supporters of open source should adopt the more positive image of inheritance, and needn't frighten people into thinking that they can be accidentally infected by the GPL.

---

Legal advice must be provided in the course of an attorney-client relationship specifically with reference to all the facts of a particular situation. Even though an attorney wrote this article, the information in this article must not be relied upon as a substitute for obtaining specific legal advice from a licensed attorney.