The Problem Is Software Controlled by Its Developer

GNU philosophy

This article was first published in the March/April 2008 issue of http://bostonreview.net and is a response to Jonathan Zittrain's "Protecting the Internet without Wrecking It," which was published in the same issue and is available at http://bostonreview.net/BR33.2/zittrain.php.

This document is part of GNU philosophy, the GNU Project's exhaustive collection of articles and essays about free software and related matters.

The Problem Is Software Controlled by Its Developer

I fully agree with Jonathan Zittrain's conclusion that we should not abandon general-purpose computers. Alas, I disagree completely with the path that led him to it. He presents serious security problems as an intolerable crisis, but I'm not convinced. Then he forecasts that users will panic in response and stampede toward restricted computers (which he calls "appliances"), but there is no sign of this happening.

Zombie machines are a problem, but not a catastrophe. Moreover, far from panicking, most users ignore the issue. Today, people are indeed concerned about the danger of phishing (mail and web pages that solicit personal information for fraud), but using a browsing-only device instead of a general computer won't protect you from that.

Meanwhile, Apple has reported that 25 percent of iPhones have been unlocked. Surely at least as many users would have preferred an unlocked iPhone but were afraid to try a forbidden recipe to obtain it. This refutes the idea that users generally prefer that their devices be locked.

It is true that a general computer lets you run programs designed to spy on you, restrict you, or even let the developer attack you. Such programs include KaZaA, RealPlayer, Adobe Flash, Windows Media Player, Microsoft Windows, and MacOS. Windows Vista does all three of those things; it also lets Microsoft change the software without asking, or command it to permanently cease normal functioning.

But restricted computers are no help, because they present the same problem for the same reason.

The iPhone is designed for remote attack by Apple. When Apple remotely destroys iPhones that users have unlocked to enable other uses, that is no better than when Microsoft remotely sabotages Vista. The TiVo is designed to enforce restrictions on access to the recordings you make, and reports what you watch. E-book readers such as the Amazon "Swindle" are designed to stop you from sharing and lending your books. Features that artificially obstruct use of your data are known as Digital Restrictions Management (DRM); our protest campaign against DRM is hosted at http://defectivebydesign.org. (Our adversaries call DRM "Digital Rights Management" based on their idea that restricting you is their right. When you choose a term, you choose your side.)

The nastiest of the common restricted devices are cell phones. They transmit signals for tracking your whereabouts even when switched "off"; the only way to stop this is to take out all the batteries. Many can also be turned on remotely, for listening, unbeknownst to you. (The FBI is already taking advantage of this feature, and the US Commerce Department lists this danger in its Security Guide.) Cellular phone network companies regularly install software in users phones, without asking, to impose new usage restrictions.

With a general computer you can escape by rejecting such programs. You don't have to have KaZaA, RealPlayer, Adobe Flash, Windows Media Player, Microsoft Windows or MacOS on your computer (I don't). By contrast, a restricted computer gives you no escape from the software built into it.

The root of this problem, both in general PCs and restricted computers, is software controlled by its developer. The developer (typically a corporation) controls what the

program does, and prevents everyone else from changing it. If the developer decides to put in malicious features, even a master programmer cannot easily remove them.

The remedy is to give the users more control, not less. We must insist on free/libre software, software that the users are free to change and redistribute. Free/libre software develops under the control of its users: if they don't like its features, for whatever reason, they can change them. If you're not a programmer, you still get the benefit of control by the users. A programmer can make the improvements you would like, and publish the changed version. Then you can use it too.

With free/libre software, no one has the power to make a malicious feature stick. Since the source code is available to the users, millions of programmers are in a position to spot and remove the malicious feature and release an improved version; surely someone will do it. Others can then compare the two versions to verify independently which version treats users right. As a practical fact, free software is generally free of designed-in malware.

Many people do acquire restricted devices, but not for motives of security. Why do people choose them?

Sometimes it is because the restricted devices are physically smaller. I edit text all day (literally) and I find the keyboard and screen of a laptop well worth the size and weight. However, people who use computers differently may prefer something that fits in a pocket. In the past, these devices have typically been restricted, but they weren't chosen for that reason.

Now they are becoming less restricted. In fact, the OpenMoko cell phone features a main computer running entirely free/libre software, including the GNU/Linux operating system normally used on PCs and servers.

A major cause for the purchase of some restricted computers is financial sleight of hand. Game consoles, and the iPhone, are sold for an unsustainably low price, and the manufacturers subsequently charge when you use them. Thus, game developers must pay the game console manufacturer to distribute a game, and they pass this cost on to the user. Likewise, AT&T pays Apple when an iPhone is used as a telephone. The low up-front price misleads customers into thinking they will save money.

If we are concerned about the spread of restricted computers, we should tackle the issue of the price deception that sells them. If we are concerned about malware, we should insist on free software that gives the users control.

Postnote

Zittrain's suggestion to reduce the statute of limitations on software patent lawsuits is a tiny step in the right direction, but it is much easier to solve the whole problem. Software patents are an unnecessary, artificial danger imposed on all software developers and users in the US. Every program is a combination of many methods and techniques—thousands of them in a large program. If patenting these methods is allowed, then hundreds of those used in a given program are probably patented. (Avoiding them is not feasible; there may be no alternatives, or the alternatives may be patented too.) So the developers of the program face hundreds of potential lawsuits from parties unknown, and the users can be sued as well.

The complete, simple solution is to eliminate patents from the field of software. Since the patent system is created by statute, eliminating patents from software will be easy given sufficient political will. (See http://www.endsoftpatents.org.)