Can You Trust Your Computer?

GNU philosophy

This essay was first published on http://gnu.org, in 2002.

Can You Trust Your Computer?

Who should your computer take its orders from? Most people think their computers should obey them, not obey someone else. With a plan they call "trusted computing," large media corporations (including the movie companies and record companies), together with computer companies such as Microsoft and Intel, are planning to make your computer obey them instead of you. (Microsoft's version of this scheme is called Palladium.) Proprietary programs have included malicious features before, but this plan would make it universal.

Proprietary software means, fundamentally, that you don't control what it does; you can't study the source code, or change it. It's not surprising that clever businessmen find ways to use their control to put you at a disadvantage. Microsoft has done this several times: one version of Windows was designed to report to Microsoft all the software on your hard disk; a recent "security" upgrade in Windows Media Player required users to agree to new restrictions. But Microsoft is not alone: the KaZaA music-sharing software is designed so that KaZaA's business partner can rent out the use of your computer to its clients. These malicious features are often secret, but even once you know about them it is hard to remove them, since you don't have the source code.

In the past, these were isolated incidents. "Trusted computing" would make the practice pervasive. "Treacherous computing" is a more appropriate name, because the plan is designed to make sure your computer will systematically disobey you. In fact, it is designed to stop your computer from functioning as a general-purpose computer. Every operation may require explicit permission.

The technical idea underlying treacherous computing is that the computer includes a digital encryption and signature device, and the keys are kept secret from you. Proprietary programs will use this device to control which other programs you can run, which documents or data you can access, and what programs you can pass them to. These programs will continually download new authorization rules through the Internet, and impose those rules automatically on your work. If you don't allow your computer to obtain the new rules periodically from the Internet, some capabilities will automatically cease to function.

Of course, Hollywood and the record companies plan to use treacherous computing for Digital Restrictions Management (DRM), so that downloaded videos and music can be played only on one specified computer. Sharing will be entirely impossible, at least using the authorized files that you would get from those companies. You, the public, ought to have both the freedom and the ability to share these things. (I expect that someone will find a way to produce unencrypted versions, and to upload and share them, so DRM will not entirely succeed, but that is no excuse for the system.)

Making sharing impossible is bad enough, but it gets worse. There are plans to use the same facility for email and documents—resulting in email that disappears in two weeks, or documents that can only be read on the computers in one company.

Imagine if you get an email from your boss telling you to do something that you think is risky; a month later, when it backfires, you can't use the email to show that the decision was not yours. "Getting it in writing" doesn't protect you when the order is written in disappearing ink.

Imagine if you get an email from your boss stating a policy that is illegal or morally outrageous, such as to shred your company's audit documents, or to allow a dangerous

threat to your country to move forward unchecked. Today you can send this to a reporter and expose the activity. With treacherous computing, the reporter won't be able to read the document; her computer will refuse to obey her. Treacherous computing becomes a paradise for corruption.

Word processors such as Microsoft Word could use treacherous computing when they save your documents, to make sure no competing word processors can read them. Today we must figure out the secrets of Word format by laborious experiments in order to make free word processors read Word documents. If Word encrypts documents using treacherous computing when saving them, the free software community won't have a chance of developing software to read them—and if we could, such programs might even be forbidden by the Digital Millennium Copyright Act.

Programs that use treacherous computing will continually download new authorization rules through the Internet, and impose those rules automatically on your work. If Microsoft, or the US government, does not like what you said in a document you wrote, they could post new instructions telling all computers to refuse to let anyone read that document. Each computer would obey when it downloads the new instructions. Your writing would be subject to 1984-style retroactive erasure. You might be unable to read it yourself.

You might think you can find out what nasty things a treacherous-computing application does, study how painful they are, and decide whether to accept them. Even if you can find this out, it would be foolish to accept the deal, but you can't even expect the deal to stand still. Once you come to depend on using the program, you are hooked and they know it; then they can change the deal. Some applications will automatically download upgrades that will do something different—and they won't give you a choice about whether to upgrade.

Today you can avoid being restricted by proprietary software by not using it. If you run GNU/Linux or another free operating system, and if you avoid installing proprietary applications on it, then you are in charge of what your computer does. If a free program has a malicious feature, other developers in the community will take it out, and you can use the corrected version. You can also run free application programs and tools on nonfree operating systems; this falls short of fully giving you freedom, but many users do it.

Treacherous computing puts the existence of free operating systems and free applications at risk, because you may not be able to run them at all. Some versions of treacherous computing would require the operating system to be specifically authorized by a particular company. Free operating systems could not be installed. Some versions of treacherous computing would require every program to be specifically authorized by the operating system developer. You could not run free applications on such a system. If you did figure out how, and told someone, that could be a crime.

There are proposals already for US laws that would require all computers to support treacherous computing, and to prohibit connecting old computers to the Internet. The CBDTPA (we call it the Consume But Don't Try Programming Act) is one of them. But even if they don't legally force you to switch to treacherous computing, the pressure to accept it may be enormous. Today people often use Word format for communication, although this causes several sorts of problems (see *We Can Put an End to Word Attachments*). If only a treacherous-computing machine can read the latest Word documents, many people will switch to it, if they view the situation only in terms of individual action (take it or leave

it). To oppose treacherous computing, we must join together and confront the situation as a collective choice.

For further information about treacherous computing, see http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html.

To block treacherous computing will require large numbers of citizens to organize. We need your help! Please support Defective by Design, the FSF's campaign against Digital Restrictions Management.

Postscripts

1. The computer security field uses the term "trusted computing" in a different way—beware of confusion between the two meanings.
2. The GNU Project distributes the GNU Privacy Guard, a program that implements public-key encryption and digital signatures, which you can use to send secure and private email. It is useful to explore how GPG differs from treacherous computing, and see what makes one helpful and the other so dangerous.

   When someone uses GPG to send you an encrypted document, and you use GPG to decode it, the result is an unencrypted document that you can read, forward, copy, and even reencrypt to send it securely to someone else. A treacherous-computing application would let you read the words on the screen, but would not let you produce an unencrypted document that you could use in other ways. GPG, a free software package, makes security features available to the users; *they* use *it.* Treacherous computing is designed to impose restrictions on the users; *it* uses *them.*
3. The supporters of treacherous computing focus their discourse on its beneficial uses. What they say is often correct, just not important.

   Like most hardware, treacherous-computing hardware can be used for purposes which are not harmful. But these features can be implemented in other ways, without treacherous-computing hardware. The principal difference that treacherous computing makes for users is the nasty consequence: rigging your computer to work against you.

   What they say is true, and what I say is true. Put them together and what do you get? Treacherous computing is a plan to take away our freedom, while offering minor benefits to distract us from what we would lose.
4. Microsoft presents Palladium as a security measure, and claims that it will protect against viruses, but this claim is evidently false. A presentation by Microsoft Research in October 2002 stated that one of the specifications of Palladium is that existing operating systems and applications will continue to run; therefore, viruses will continue to be able to do all the things that they can do today.

   When Microsoft employees speak of "security" in connection with Palladium, they do not mean what we normally mean by that word: protecting your machine from things you do not want. They mean protecting your copies of data on your machine from access by you in ways others do not want. A slide in the presentation listed several types of secrets Palladium could be used to keep, including "third party secrets" and "user secrets"—but it put "user secrets" in quotation marks, recognizing that this is somewhat of an absurdity in the context of Palladium.

The presentation made frequent use of other terms that we frequently associate with the context of security, such as "attack," "malicious code," "spoofing," as well as "trusted." None of them means what it normally means. "Attack" doesn't mean someone trying to hurt you, it means you trying to copy music. "Malicious code" means code installed by you to do what someone else doesn't want your machine to do. "Spoofing" doesn't mean someone's fooling you, it means your fooling Palladium. And so on.

5. A previous statement by the Palladium developers stated the basic premise that whoever developed or collected information should have total control of how you use it. This would represent a revolutionary overturn of past ideas of ethics and of the legal system, and create an unprecedented system of control. The specific problems of these systems are no accident; they result from the basic goal. It is the goal we must reject.