The Word "Hacker"

April 2004

To the popular press, "hacker" means someone who breaks into computers. Among programmers it means a good programmer. But the two meanings are connected. To programmers, "hacker" connotes mastery in the most literal sense: someone who can make a computer do what he wants—whether the computer wants to or not.

To add to the confusion, the noun "hack" also has two senses. It can be either a compliment or an insult. It's called a hack when you do something in an ugly way. But when you do something so clever that you somehow beat the system, that's also called a hack. The word is used more often in the former than the latter sense, probably because ugly solutions are more common than brilliant ones.

Believe it or not, the two senses of "hack" are also connected. Ugly and imaginative solutions have something in common: they both break the rules. And there is a gradual continuum between rule breaking that's merely ugly (using duct tape to attach something to your bike) and rule breaking that is brilliantly imaginative (discarding Euclidean space).

Hacking predates computers. When he was working on the Manhattan Project, Richard Feynman used to amuse himself by breaking into safes containing secret documents. This tradition continues today. When we were in grad school, a hacker friend of mine who spent too much time around MIT had his own lock picking kit. (He now runs a hedge fund, a not unrelated enterprise.)

It is sometimes hard to explain to authorities why one would want to do such things. Another friend of mine once got in trouble with the government for breaking into computers. This had only recently been declared a crime, and the FBI found that their usual investigative technique didn't work. Police investigation apparently begins with a motive. The usual motives are few: drugs, money, sex, revenge. Intellectual curiosity was not one of the motives on the FBI's list. Indeed, the whole concept seemed foreign to them.

Those in authority tend to be annoyed by hackers' general attitude of disobedience. But that disobedience is a byproduct of the qualities that make them good programmers. They may laugh at the CEO when he talks in generic corporate newspeech, but they also laugh at someone who tells them a certain problem can't be solved. Suppress one, and you suppress the other.

This attitude is sometimes affected. Sometimes young programmers notice the eccentricities of eminent hackers and decide to adopt some of their own in order to seem smarter. The fake version is not merely annoying; the prickly attitude of these posers can actually slow the process of innovation.

But even factoring in their annoying eccentricities, the disobedient attitude of hackers is a net win. I wish its advantages were better understood.

For example, I suspect people in Hollywood are simply mystified by hackers' attitudes toward copyrights. They are a perennial topic of heated discussion on Slashdot. But why should people who program computers be so concerned about copyrights, of all things?

Partly because some companies use *mechanisms* to prevent copying. Show any hacker a lock and his first thought is how to pick it. But there is a deeper reason that hackers are alarmed by measures like copyrights and patents. They see increasingly aggressive measures to protect "intellectual property" as a threat to the intellectual freedom they need to do their job. And they are right.

It is by poking about inside current technology that hackers get ideas for the next generation. No thanks, intellectual homeowners may say, we don't need any outside help. But they're wrong. The next generation of computer technology has often—perhaps more often than not—been developed by outsiders.

In 1977 there was no doubt some group within IBM developing what they expected to be the next generation of business computer. They were mistaken. The next generation of business computer was being developed on entirely different lines by two long-haired guys called Steve in a garage in Los Altos. At about the same time, the powers that be were cooperating to develop the official next generation operating system, Multics. But two guys who thought Multics excessively complex went off and wrote their own. They gave it a name that was a joking reference to Multics: Unix.

The latest intellectual property laws impose unprecedented restrictions on the sort of poking around that leads to new ideas. In the past, a competitor might use patents to prevent you from selling a copy of something they made, but they couldn't prevent you from taking one apart to see how it worked. The latest laws make this a crime. How are we to develop new technology if we can't study current technology to figure out how to improve it?

Ironically, hackers have brought this on themselves. Computers

are responsible for the problem. The control systems inside machines used to be physical: gears and levers and cams. Increasingly, the brains (and thus the value) of products is in software. And by this I mean software in the general sense: i.e. data. A song on an LP is physically stamped into the plastic. A song on an iPod's disk is merely stored on it.

Data is by definition easy to copy. And the Internet makes copies easy to distribute. So it is no wonder companies are afraid. But, as so often happens, fear has clouded their judgement. The government has responded with draconian laws to protect intellectual property. They probably mean well. But they may not realize that such laws will do more harm than good.

Why are programmers so violently opposed to these laws? If I were a legislator, I'd be interested in this mystery—for the same reason that, if I were a farmer and suddenly heard a lot of squawking coming from my hen house one night, I'd want to go out and investigate. Hackers are not stupid, and unanimity is very rare in this world. So if they're all squawking, perhaps there is something amiss.

Could it be that such laws, though intended to protect America, will actually harm it? Think about it. There is something very *American* about Feynman breaking into safes during the Manhattan Project. It's hard to imagine the authorities having a sense of humor about such things over in Germany at that time. Maybe it's not a coincidence.

Hackers are unruly. That is the essence of hacking. And it is also the essence of Americanness. It is no accident that Silicon Valley is in America, and not France, or Germany, or England, or Japan. In those countries, people color inside the lines.

I lived for a while in Florence. But after I'd been there a few months I realized that what I'd been unconsciously hoping to find there was back in the place I'd just left. The reason Florence is famous is that in 1450, it was New York. In 1450 it was filled with the kind of turbulent and ambitious people you find now in America. (So I went back to America.)

It is greatly to America's advantage that it is a congenial atmosphere for the right sort of unruliness—that it is a home not just for the smart, but for smart-alecks. And hackers are invariably smart-alecks. If we had a national holiday, it would be April 1st. It says a great deal about our work that we use the same word for a brilliant or a horribly cheesy solution. When we cook one up we're not always 100% sure which kind it is. But as long as it has the right sort of wrongness, that's a promising sign. It's odd that people think of programming as precise and methodical. *Computers* are precise and methodical. Hacking is something you do with a gleeful laugh.

In our world some of the most characteristic solutions are not far removed from practical jokes. IBM was no doubt rather surprised by the consequences of the licensing deal for DOS, just as the hypothetical "adversary" must be when Michael Rabin solves a

problem by redefining it as one that's easier to solve.

Smart-alecks have to develop a keen sense of how much they can get away with. And lately hackers have sensed a change in the atmosphere. Lately hackerliness seems rather frowned upon.

To hackers the recent contraction in civil liberties seems especially ominous. That must also mystify outsiders. Why should we care especially about civil liberties? Why programmers, more than dentists or salesmen or landscapers?

Let me put the case in terms a government official would appreciate. Civil liberties are not just an ornament, or a quaint American tradition. Civil liberties make countries rich. If you made a graph of GNP per capita vs. civil liberties, you'd notice a definite trend. Could civil liberties really be a cause, rather than just an effect? I think so. I think a society in which people can do and say what they want will also tend to be one in which the most efficient solutions win, rather than those sponsored by the most influential people. Authoritarian countries become corrupt; corrupt countries become poor; and poor countries are weak. It seems to me there is a Laffer curve for government power, just as for tax revenues. At least, it seems likely enough that it would be stupid to try the experiment and find out. Unlike high tax rates, you can't repeal totalitarianism if it turns out to be a mistake.

This is why hackers worry. The government spying on people doesn't literally make programmers write worse code. It just leads eventually to a world in which bad ideas win. And because this is so important to hackers, they're especially sensitive to it. They can sense totalitarianism approaching from a distance, as animals can sense an approaching thunderstorm.

It would be ironic if, as hackers fear, recent measures intended to protect national security and intellectual property turned out to be a missile aimed right at what makes America successful. But it would not be the first time that measures taken in an atmosphere of panic had the opposite of the intended effect.

There is such a thing as Americanness. There's nothing like living abroad to teach you that. And if you want to know whether something will nurture or squash this quality, it would be hard to find a better focus group than hackers, because they come closest of any group I know to embodying it. Closer, probably, than the men running our government, who for all their talk of patriotism remind me more of Richelieu or Mazarin than Thomas Jefferson or George Washington.

When you read what the founding fathers had to say for themselves, they sound more like hackers. "The spirit of resistance to government," Jefferson wrote, "is so valuable on certain occasions, that I wish it always to be kept alive."

Imagine an American president saying that today. Like the remarks of an outspoken old grandmother, the sayings of the founding fathers have embarrassed generations of their less confident successors. They remind us where we come from. They

remind us that it is the people who break rules that are the source of America's wealth and power.

Those in a position to impose rules naturally want them to be obeyed. But be careful what you ask for. You might get it.

Thanks to Ken Anderson, Trevor Blackwell, Daniel Giffin, Sarah Harlin, Shiro Kawai, Jessica Livingston, Matz, Jackie McDonough, Robert Morris, Eric Raymond, Guido van Rossum, David Weinberger, and Steven Wolfram for reading drafts of this essay.

(The image shows Steves Jobs and Wozniak with a "blue box." Photo by Margret Wozniak. Reproduced by permission of Steve Wozniak.)

- Portuguese Translation
- Hebrew Translation
- Romanian Translation
