DOI:10.1145/1435417.1435426

Jonathan Zittrain

Law and Technology
The End of the Generative Internet

Exploring the expectations and implications for version 2.0 of the Net's new gated communities.

I RECENTLY WROTE a book about the future of the Internet. The book's thesis is that the mainstream computing environment we've experienced for the past 30-plus years—dating from the introduction of the first mainstream personal computer, the Apple II, in 1977—is an anomaly. The basic building blocks of modern IT are PCs that anyone can reprogram, connected to an Internet that unquestioningly routes bits between two arbitrary points. This has led to a generative revolution where novel and disruptive technologies have come from obscure backwaters—and conquered. While incumbents bet on (or were) gated-community networks like CompuServe, Prodigy, and AOL, or makers of "smart appliances" such as dedicated word processors and video-game consoles, dark-horse candidates like the Internet and the PC emerged unexpectedly and triumphed, helped along by commercial forces that belatedly hopped on their bandwagons.

So why anomalous? Technologies that allow—indeed, depend on—contributions from anywhere and trust of unknown contributors at first thrive in elite communities where people have the technical astuteness to know how to manage them and the ethos of tinkering and mutual assistance. As the platforms' popularity increases and they enter the mainstream, greater numbers and a decline in the average user's skill set make them increasingly worth subverting. For example, spam is more profitable when there are hundreds of millions of recipients instead of thousands, and worms and Trojans have that much more personal data to mine or processors to hijack when over one billion PCs are in use.[1]

Imagine if Microsoft had adjusted Windows to act the way the iPhone and Facebook apps platforms do.

One approach to the problem is to try to double-click our way out of it: to subscribe to antivirus software that tries to stop harmful code from running on our machines. Such Patriot missile-style defense aspires to keep bad code as a mere nuisance, but it depends on 100% accuracy and a willingness by users to defer to the recommendations of their antivirus packages—difficult when there are plenty of false positives.

Another approach is to eliminate or qualify the generative character of our technology. Apple's iPhone was introduced in 2007, and its first version, like its near-twin iPod, allowed no outside code at all. As the book went to press in October 2007, Apple, Inc., provided perfect bookends for the trajectory of the consumer information technology ecosystem: "Rather than a platform that invites innovation, the iPhone comes preprogrammed. You are not allowed to add programs to the all-in-one device that Steve Jobs sells you.

ILLUSTRATION BY JOHN HERSEY

Its functionality is locked in, though Apple can change it through remote updates. Indeed, to those who managed to tinker with the code to enable the iPhone to support more or different applications, Apple threatened (and then delivered on the threat) to transform the iPhone into an iBrick. The machine was not to be generative beyond the innovations that Apple (and its exclusive carrier, AT&T) wanted. Whereas the world would innovate for the Apple II, only Apple would innovate for the iPhone. (A promised software development kit may allow others to program the iPhone with Apple's permission.)"[3]

The parenthetical remark has since come true (the iPhone Software Development Kit was released in March 2008), and as of this writing in late 2008, several months after the introduction of iPhone 2.0, it is difficult to imagine the iPhone without its outside apps. This is a partial vindication of generative technologies over sterile ones, but with caveats that may make this the worst rather than the best of both worlds.

The application environment for the iPhone flips from the PC-with-antivirus-software's blacklist system to a whitelist scheme unchangeable by the user. With rare exceptions, such as a special ad hoc program allowing very limited distribution. Outside developers must register with Apple, promise not to disclose anything about how apps are written, and if approved they may then submit their software to Apple for review and possible inclusion in the iPhone App Store, the only way for the public at large to obtain it.

Apple can change its mind at any time about a particular piece of software's inclusion in its store. And even software already obtained from the iPhone App Store can be recalled—that's just a subset of Apple's ability to remotely reprogram any aspect of the phone at any time.

For vendors of iPhone apps, Apple's goodwill is thus vital. Apart from deciding whether an app lives or dies, Apple can feature favored apps in its store, and it can make app updates and bug fixes available slowly or quickly. For these reasons a gag order in the license agreement demanding that software authors not discuss codewriting for unreleased software was taken very seriously among app developers. Apple doesn't need to bring a lawsuit against a developer who violates license terms; it already has the power to destroy the iPhone-based livelihood of anyone disfavored, for any reason.

NetShare, an app that allowed users to gain wireless connectivity for their PCs through a connected iPhone, disappeared, and as of this writing the company says it doesn't know why. Box Office, an app that provided movie times, was removed from the iPhone App Store for several weeks. Its developer has declined any comment since its return. Other developers for mail and podcast-related programs say their apps were turned down with the explanation they were "duplicative" of (that is, competitive with) existing iPhone functionality.

Should we care? Apple likely wouldn't kill apps people really like since they make money along with the authors: 30% of all sales. And people think of an iPhone as a more unified device, expecting all of it to work at high quality, so gatekeeping might help keep malicious or poor-quality apps away.

On the other hand, people don't know what they're missing—and firms can be very ineffective, despite their own economic interests, in recognizing the value of truly novel contributions from outsiders that might take a while to catch on. Who would have invested in Wikipedia at the beginning? And if Wikipedia required an incumbent gatekeeper's approval or permission to get started, it might have failed to receive it—or languished at the bottom of a to-do list among hundreds of other apps and services awaiting review.

This phenomenon isn't exclusive to Apple, of course. Even today's PCs have a flavor of it. Microsoft offers a monthly malicious software removal tool, which unobtrusively goes through a PC to remove malware. Presumably it would become much less popular if Microsoft, or someone regulating Microsoft, tried to use the tool to remove software that people liked; no one seems to have tried to get Microsoft to kill anything yet, though, and such attempts are limited since any new app can immediately be installed on a PC—including one that shuts down a Microsoft app-removal tool.

Back on the whitelisting side of the spectrum, Web development platforms like Facebook Apps have restrictions that essentially mirror those of the iPhone. And when Facebook kills an app, the app is naturally not only unavailable to new users, but disabled for current ones, too. So Superwall or Secret Crush can go from millions of users to zero in a heartbeat. People learn about new apps through their friends' Facebook newsfeeds—and Facebook can adjust just how much news an app will generate there. A Great Apps program allows Facebook to pick winners and feature them more openly, even as some developers grumble that the functionalities they build are sometimes incorporated into apps written by Facebook itself[2]—and then effortlessly promoted more than the outsiders' original.

Its modest malicious software removal tool aside, imagine if Microsoft had adjusted Windows to act the way the iPhone and Facebook apps platforms do.

The iPhone apps model is powerful, and it is serving some useful purpose in shielding people, prospectively and retroactively, against bad code.

WordPerfect would owe Microsoft 30% on sales of every copy of its word processor—if it sold any, since Word could be featured by Microsoft to its users much more readily, or rejected entirely as duplicative of Word. (Recall that a main basis for the Microsoft antitrust case in the 1990s arose from Microsoft's attempts to force PC sellers to include the Internet Explorer browser on their PCs' desktops out of the box. The ways in which Facebook or Apple can feature their own apps over those of others dwarf that.) Of course, Microsoft could change that percentage owed at any time—or make it a flat fee. The makers of, say, Quicken, could find that they owe 70% or 80% on every app, take it or leave it. If they leave it, Quicken would stop working on every PC on which it had previously been installed.

And anyone objecting to an app—say, the movie and music industries beholding the rise of Kazaa or BitTorrent—could pressure Microsoft to kill it the day it appeared. We recently experienced this scenario when Hasbro, owner of the intellectual property rights to Scrabble in the U.S. and Canada, pressured Facebook to kill Scrabulous, a Scrabble knockoff. No court needed to weigh in on this decision.

We likely wouldn't accept this situation in PC architecture, and yet it is commonplace in the ecosystem that will soon replace it. Is the difference that Microsoft had overwhelming market share—an acknowledged monopoly—over PCs? That certainly counts, but even if one vendor doesn't capture the mobile phone or social networking spaces, the choices among them are shaping up to be choices among gated communities: equivalent to the old AOL vs. Prodigy vs. CompuServe, with the Internet not in the running. This is one reason why Google's Android project is so fascinating: an attempt to bring the generativity of the PC to the mobile phone space. Without a security model better than the PC's security provisions, however, Android is a tough proposition. How long will users tolerate a phone for which clicking on the wrong link can disable it?

The iPhone apps model is powerful, and it is serving some useful purpose in shielding people, prospectively and retroactively, against bad code. It is so powerful and popular that we will see it extended to PC-like platforms, too, with the 30-year run of open season for new software drawing to a close.

The way forward—for both PCs and smartphones—lies in a new security architecture that lets users make better-informed decisions about whether to run new software. We could aggregate data and make it freely available—how many experts have installed this same code? On average, what impact does the code have on the environment in which it runs, as measured by crashes or pop-ups or user satisfaction? A user deciding whether to run new code could use that data to make a simple decision, instead of letting the autocratic voice of preprogrammed security software dictate the result. Such an architecture would be more flexible than what we currently use. There are many details to work out, but without ways of managing our generative platforms without a central gatekeeper, chances seem strong that most people will accept—even demand—outside control. C

References

1. Computers in use pass 1 billion mark: Gartner. Reuters (June 23, 2008); www.reuters.com/article/technologyNews/idUSL2324525420080623.
2. Stone, B. New tool from Facebook extends its Web presence. *New York Times* (July 24, 2008); www.nytimes.com/2008/07/24/technology/24facebook.html.
3. Zittrain, J. *The Future of the Internet—And How to Stop It.* Yale University Press, Penguin U.K., and Creative Commons, 2008; www.futureoftheinternet.org/download.

Jonathan Zittrain (zittrain@law.harvard.edu) is Professor of Law at Harvard Law School in Cambridge, MA.