Fumbling the Future, Again

Good security must be built in at the start. Doing otherwise creates abstractions and interfaces that are insecure but cannot be changed because customers are already using them. This does not bode well for achieving security in clouds that offer existing interfaces (such as an instruction set, an OS, or an extant application). But it might not be too late to build secure cell phones. So the relatively low level of interest in cell phone security by industry, government, and researchers is dismaying. We are about to fumble the future. Again.

Builders of each new generation of computing platforms seem to ignore security. Why should this be? We are often told that implementing security would only further stress scarce resources. So developers leave security out, awaiting Moore's law to work its magic.

In addition, security inevitably inconveniences users, imposes restrictions on information sharing, and disciplines component interactions. Such a runtime is seen as being antithetical to a developer's goal of creating platforms usable by many and in unanticipated ways. Moreover, unable to anticipate what their platforms will need to protect, developers have no way to gauge how much security to provide. So what gets deployed is basically an attractive nuisance: it attracts users and puts at risk value that is stored on the platform and possibly elsewhere.

Support for security was and could be minimal in early cell phones because the platform was closed, and the value being hosted was relatively low. Today, processing capacity is present to host arbitrary applications. We have thus reached a technology inflection point. With more than 4 billion cell phones in the world, with growth that exceeds that of desktops and laptops, and with a total cost of ownership that makes cell phones affordable to more of the world than a PC will ever be, the market is responding. And hundreds of thousands of applications are available today for download onto cell phones.

As with PCs, cell phone use is limited mostly by the choice of which extensions have been installed (although the cell phone's small screen and keyboard does preclude certain applications). Portability, in addition, makes cell phones an ideal platform for performing authentication—"something you have" is buttressed by an interface for "something you know." So cell phones can be used to provide access to real objects (such as buildings) as well as to electronic objects (such as bank accounts) with significant value. However, by being extensible and interactive like desktop PCs, the potential exists for the same kinds of attacks against cell phones that we see against the desktop PC.

Improved cell phone security will likely involve technology and policy. For example, a cell phone OS must enforce strong isolation between applications, yet allow sharing of physical resources (like the user interface) as well as logical resources (so applications can communicate). In short, new system software is needed that corrects insecure designs found in the OSs that run today's desktop PCs.

Policy could also help promote cell phone security, for example, by shunning the software-ownership model that has evolved for PCs. In the past, cell phone providers (and not customers) took full responsibility for the software on their customers' phones. This allowed providers to eliminate a bug or vulnerability soon after it was identified. In addition, some cell phone producers today tightly control what applications they allow to run on their customers' platforms. Matters of censorship and monopoly-power aside (because they can be dealt with by legislation), there are decided security advantages to this provider-based software ownership model—it could, for example, allow third-party security and privacy validation of applications, and it could also enable accountability.

Forewarned is forearmed. Cell phone security needs far more attention than it is getting. We don't have to fumble the future, again. □

Fred B. Schneider
Associate Editor in Chief