The ActiveX Security Model

by M. E. Kabay, PhD, CISSP
Security Leader
INFOSEC Group
AtomicTangerine, Inc.

According to Microsoft < http://www.microsoft.com/com/tech/ActiveX.asp >, "ActiveX® controls are among the many types of components that use COM [Component Object Model; see < http://www.microsoft.com/com/default.asp > for an overview] technologies to provide interoperability with other types of COM components and services. ActiveX controls are the third version of OLE controls (OCX), providing a number of enhancements specifically designed to facilitate distribution of components over high-latency networks and to provide integration of controls into Web browsers. These enhancements include features such as incremental rendering and code signing, to allow users to identify the authors of controls before allowing them to execute."

The security model for ActiveX programs, called "controls," running on a Windows 9x system permits signed code to run with no constraints. ActiveX controls can call any system routines. For example, the "Exploder" demonstration control was written by Fred McLain and is described on < http://www.halcyon.com/mclain/ActiveX/welcome.html >. McLain explains:

> Exploder performs a clean shutdown of Windows 95 from a web page. On "Green Machines", particularly those with a power conservative BIOS, (mostly laptop computers) it also turns the power off after shutdown. For the technical folks out there, this is a call to the windows API function ExitWindowsEx() with the flags EWX_SHUTDOWN and EWX_POWEROFF set. For the less technical, it's the same thing as the "Shut Down" menu item on the "Start" button, but with the power off feature added. <

For a more extensive discussion of Exploder and how Microsoft and Verisign responded to its publication, see its author's FAQ at < http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm >.

The fundamental problem with the security model for ActiveX is the weak relation between knowing the stated origin of code and knowing that the code is trustworthy. Sometimes the correlation between origin and quality is strong; for example, we may know very well that if a particular student in a programming class has written a particular piece of code, that code may be poorly written. For that matter, we may know that if a particular company produces code, the code will almost certainly be poorly written. However, in theory, a Bad Person could obtain a certificate and sign the Dangerous_Code control before using it on a Web page. How, exactly, would a visitor to the Web page determine whether to allow Dangerous_Code to download and run? On what basis would a user -- especially a normal user with no idea of how ActiveX or any other code should be designed -- judge whether to allow the Dangerous_Code control to execute?

In your experience as network managers and users, how many non-technical users ever even _think_ of verifying the nature of the person or organization that signed a control before downloading and executing it in their browser session? And if they did want to look into who

wrote (or signed) the code, on what basis would they judge the code's fitness?

No, I'm very sorry to conclude that claiming that certificates of origin necessarily tell us about the quality and safety of code is a non sequitur: the second element does not follow from the first.

In practical terms, I do not allow ActiveX to execute at all in my browsers unless I feel that I absolutely have to for a sound business purpose. And I recommend to Web designers that they ensure that visitors who disallow ActiveX will nonetheless be able to obtain a reasonable amount of information without using this seriously flawed conception of software security.

* * *

Mich Kabay can be reached by e-mail at <mkabay@atomictangerine.com>. He invites inquiries about a wide range of information security courses and INFOSEC consulting services that he and his colleagues at AtomicTangerine would be delighted to deliver to your employees at your site and at your convenience. For Web-based or CD-ROM online training in security from our INFOSEC University project, see < http://infosecu.com >.

For information about AtomicTangerine, visit < http://www.atomictangerine.com >.