Thinking inside the box

devblogs.microsoft.com/oldnewthing/20091019-00

October 19, 2009

Raymond Chen

Commenter Nick asks whether any though has been given to running applications in a sandbox where they are given access only to their installation directory, My Documents, and a handful of other directories and registry keys. “I feel that this would seriously prevent viruses/spyware from being as effective, and apps would not be able to dump files all over the users’ HD.” Yes, a lot of thought has been given to sandboxing (most of which I am not at liberty to discuss) but the compatibility consequences have been a constant source of trepidation. For example, if you restrict the application just to My Documents and other selected folders, then double-clicking a file from some other location (not on the list of “allowed” locations) would result in a strange “File not found” (or maybe “Access denied”) error message when the application tried to open a file that it didn’t have access to. Redirecting file system operations has its own problems, such as applications saying that they successfully saved the file, but when you go to look in Explorer, it’s not there! To add to the complexity, the Common Criteria (the modern version of what was formerly known as C2 security) have their own rules about what behavior is permissible. For example, it is my understanding that the Common Criteria require that when access is denied to a location, the error that is returned must be “access denied”; you are not allowed to lie to an application and say “Um… yeah… I got your location right here…” Tracking an application’s shortcuts and registry entries is complicated by the fact that the application is hardly the only entity that accesses those shortcuts and registry entries. If some other application moves the shortcut to a new location or copies it, does that get added to the tracking list? If so, which tracking list? (The one for the application that created the original shortcut or the application that copied it?) Many settings are system-wide. If an application changes a system setting, and then you uninstall the application, do you restore the system setting to the value it had before the application was installed? If you say yes, then what if you had five applications all of which were changing the same system setting, and you uninstall the third one, what do you “restore” the setting to?

Windows Vista introduces a limited sort of “sandboxing” as part of a compatibility shim for applications which write to administrative locations in the file system or registry for no good reason. Those writes are redirected to a “pretend” directory or registry key, and the application (hopefully) doesn’t know any better. That’s about the extent of my knowledge of this Windows Vista feature; if you want to read more about it, you can probably hunt around MSDN. Try the phrase *file system redirection*. (This will probably also turn up features related to 32-bit emulation, so you'll have to do some more filtering.)

Raymond Chen

Follow