Why can you create a PIF file that points to something that isn't an MS-DOS program?

devblogs.microsoft.com/oldnewthing/20091112-00

November 12, 2009

Raymond Chen

James MAstros asked why it's possible to create a PIF file that refers to a program that isn't an MS-DOS program. (That's only part of the question; I addressed other parts last year.) Well, for one thing, there was indeed code to prevent you from setting PIF properties for something that isn't an MS-DOS program, so the precaution was already there. But it didn't stop anybody who was really determined to try. All you had to do was create an MS-DOS program, then create a PIF file for it, and then overwrite the MS-DOS program file with something else. Since time travel has not been invented, the PIF creator code can't retroactively go back in and say, "Well, if I knew you were going to pull this trick, I wouldn't have let you create it in the first place!" You can't even enforce it at the time the PIF file is launched, because somebody could replace the file during the split second between checking the file type and actually using it. Of course, the real question is "Why, if you create a PIF file that describes something that isn't an MS-DOS program, does it still work?" It still works because the PIF file did exactly what it was supposed to do. It created an MS-DOS virtual machine with the specified parameters and then ran the program in it. Now, back in the days when PIF files were invented, if you tried to run something that wasn't an MS-DOS program inside an MS-DOS virtual machine, all that happened was that the MS-DOS stub ran, the same thing that happened if you tried to run the program from MS-DOS. The program was treated not as a Windows program but as an MS-DOS program. Windows 95 changed that. If you tried to run a Windows application from the MS-DOS command prompt, it would run the Windows application instead of telling you, "This program cannot be run in DOS mode." This change was made for two reasons. First, the existing behavior seemed pretty stupid. You're running Windows, and if you open a command prompt and try to run a Windows program, you're told, "You need Windows to run this program." It's like one of those bizarro-land government red tape nightmares, where you go to the courthouse to file some papers in person, and the clerk at the desk says, "I'm sorry, you have to file this document in person at the courthouse." Second, it was necessary to allow 32-bit console programs to run when launched from a MS-DOS command window. Since Windows 95 used the MS-DOS prompt as its command line interface (as opposed to Windows NT which used a 32-bit command prompt), it was kind of important that you be able to run 32-bit console programs from a virtual machine. Without it, the whole idea of a console program became kind of weak.

“Yeah, we have console programs, but you can’t launch them from a console.” What happens, then, if you create PIF file that points to a 32-bit program? Well, the operating system goes to all the effort to create a virtual machine to the specifications you indicated in the PIF file. You want a particular amount of extended memory? Okay, we’ll set that up. You want a custom icon? Sure, no problem. You want it to disable DPMI memory? You got it. Once that’s all set up, the virtual MS-DOS driver says, “Okay, and set the initial CS:IP for the virtual machine to the MS-DOS EXEC call to run the program the PIF file specified.” The EXEC call executes, and the interop code kicks in and launches the 32-bit Windows program. From the virtual machine’s point of view, nothing is actually wrong; you merely did something in a really roundabout way. It’s like booking a meeting room, specifying that you would like a slide projector, that the chairs and tables be set in a particular arrangement, that everybody be provided with water and a notebook, and then putting a sign on the door that says, “This meeting has moved to Location Z.” You go to all this effort to get the conference room to be set up exactly the way you want it, and then you end up not using it. The conference center doesn’t care (as long as it still gets paid).

PIF files are like shortcuts, but with the added effort of creating an MS-DOS virtual machine. And just like with shortcuts, bad guys can choose a dangerous target and make your day miserable.

Raymond Chen

Follow