How do I get the command line of another process?

devblogs.microsoft.com/oldnewthing/20091125-00

November 25, 2009

Raymond Chen

Win32 doesn't expose a process's command line to other processes. From Win32's point of view, the command line is just a conveniently initialized parameter to the process's startup code, some data copied from the launching process to the new process and forgotten. We'll get back to the Win32 point of view a little later.

If you look around in WMI, you'll find a `Win32_Process` object, and lo and behold, it has a `CommandLine` property. Let's check it out, using the standard WMI application:

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")
Set colItems = objWMIService.ExecQuery("Select * from Win32_Process")
For Each objItem in colItems
     Wscript.Echo objItem.Name
     Wscript.Echo objItem.CommandLine
Next
```

I fully anticipate that half of my readers will stop right there. "Thanks for the script. Bye!" And they won't bother reading the analysis. "Because analysis is boring, and it'll just tell me stuff I don't want to hear. The analysis is going to tell me why this won't work, or why it's a bad idea, and that just cramps my style."

Remember that from Win32's point of view, the command line is just a string that is copied into the address space of the new process. How the launching process and the new process interpret this string is governed not by rules but by convention.

What's more, since the string is merely a "preinitialized variable", a process could in principle (and many do in practice, although usually inadvertently) write to the memory that holds the command line, in which case, if you go snooping around for it, you'll see the modified command line. There is no secret hiding place where the kernel keeps the "real original command line," any more than there is a secret hiding place where the C compiler keeps the "real original parameters to a function."

This is just another manifestation of the principle of not keeping track of information you don't need.

What does this mean for people who disregard this principle and go after the command line of another process? You have to understand what you are getting is non-authoritative information. In fact, it's worse. It's information *the application itself may have changed in order to try to fool you*, so don't use it to make important decisions.

Raymond Chen

Follow