How can I make sure my program is launched only from my helper program and no other parent?

devblogs.microsoft.com/oldnewthing/20140221-00

February 21, 2014

Raymond Chen

Say you have a collection of programs which work together. One of them is the "master" program that runs the show, and it has a bunch of "assistant" programs that it launches to accomplish various subtasks. These assistants are not meant to be run by themselves; they are meant to be run only by the master program. How do you design the assistant so that it can only be run by the master? There's nothing you can do to force the assistant to be run only by the master, since anything you do to detect the case can be faked out by an attacker. (Worst case is that they just run your program under the debugger and patch out the code that looks for the master.) So the purpose of this test is not so much to create an airtight hatchway as it is to prevent users from randomly wandering into the Program Files directory and double-clicking stuff to see what happens. The simplest way of doing this is to require a command-line parameter that the master passes to say, "Hey, it's me, the master. It's okay to do that thing you do." The command line parameter could be anything. `assistant.exe /run` say. If the command line parameter is not present, then the assistant says, "Um, please don't run this program directly. Use the master." You might decide to get really fancy and make the secret handshake super-complicated, but remember that there is no added security benefit here. The user can compromise `assistant.exe` by simply attaching a debugger to it, at which point any defensive mechanism you create can simply be disabled by a sufficiently-resourceful attacker. (And there's a class of people who will see that you put a lot of work into protecting your assistant, and that will just convince them to work harder to circumvent the protection. Because something with this much protection must certainly be very valuable!) There's also a benefit to keeping the secret handshake simple: It makes it a lot easier for you to debug the assistant program. Instead of having to set up the master and then get the master to do all the things it needs to generate the secret handshake for the assistant, you can just run your assistant directly with the magic flag, and boom, you're off and debugging.

To make it even harder to run your program by accident, you can give it an extension that is not normally executable, like `.MOD`. That way, it cannot be double-clicked, but you can still pass it to `CreateProcess` or (with some cajoling) `ShellExecuteEx`.

Raymond Chen

Follow