Solving the problem rather than answering the question: How can a non-administrator modify a registry key that requires administrator permission?

devblogs.microsoft.com/oldnewthing/20150227-00

February 27, 2015

Raymond Chen

A customer opened with a question, which the customer liaison forwarded to the product group with *High Priority*. (Because, apparently, their customer is more important than any other customer.)

> Our program needs to modify a registry key in `HKEY_LOCAL_MACHINE` even when not running as an administrator. We tried setting an entry in the registry key `HKLM\Software\Microsoft\Windows NT\CurrentVersion\AppCompat-Flags\Layers` to run our application elevated, but it didn't help. We also tried setting the entry in our application manifest to say that it requires elevation, but that didn't work either. Please provide us with a way to override UAC.

The fact that they went in and tried to enable an application compatibility setting *on their own application* means that instead of fixing their program, they are just going to keep throwing garbage on the sidewalk and hope the street sweeper will still come and clean up behind them.

Upon closer questioning, they explained that setting the manifest entry didn't work in the sense that when the user ran the program, the operating system prompted for elevation. But they wanted their program to elevate without prompting.

Okay, first of all, if any program could could elevate without prompting, then there would be no point to elevation. Every program would simply ask for secret unprompted elevation, and there would be no point to elevation in the first place. So that angle is a non-starter.

We asked them for details on the problem they are having, the problem where they think writing to HKLM is the solution. That way, we can solve the problem instead of answering the question.

> When our program is installed, it prompts the person doing the installation for the name of the server to connect to. The installer writes this information to HKLM. When a non-administrator runs the program, we want them to be able to switch to a different server. That's why we need to be able to write to HKLM.

Okay, now that we understand the scenario, we can provide a solution.

First of all, the reason why you can't write to HKLM is that it would allow a non-administrative user to change the server settings of another user. Suppose that I run the program and change the server to `http://evil-hackers.example.com`. Then I log off and wait. The next person to use the computer and run the program will connect to the hacker site instead of the real site, and now I can harvest credentials or launch a man-in-the-middle attack or otherwise do bad things.

The solution, then, is to reduce the scope of any changes a non-administrative user makes to just that user. That way, if they choose to point the program at a rogue server, they are merely attacking themselves.

- At install time, write the default server information to HKLM.
- When a user changes the server, write the new server to HKCU.
- When the program needs to decide which server to connect to:
    - Check if there is a local setting in HKCU. If so, then use it.
    - If there is no setting in HKCU, then use the setting in HKLM.
- Optional: Add an administrative option to change the default server in HKLM.

Raymond Chen

Follow