Not actually crossing the airtight hatchway: Applying per-user overrides

devblogs.microsoft.com/oldnewthing/20190801-00

August 1, 2019

Raymond Chen

We receive a number of security vulnerability reports of the form “If I write the following value into the registry at `HKEY_ CURRENT_ USER\...` , then the next time the user does X, I can do bad thing Y.”

The most common version of this is where the registry key is `HKEY_ CURRENT_ USER\ Software\ Classes\ CLSID\ ...` , because that permits you to override a system COM object with a custom COM object.

The fallacy here is hiding behind the change of pronoun in the attack description: If **I** write the following value into the registry, then the next time **the user** does X, **I** can do bad thing Y.

In reality, **I** and **the user** are the same person!

In order to write to the user’s registry, you need to be that user or an administrator. Of course, if you are an administrator, then you’re already on the other side of the airtight hatchway, and this entire exercise is pointless.

That leaves the case where the attacker is the user. In other words, the attacker is attacking himself. This is not particularly interesting. It is not a security vulnerability that users can make their own lives miserable. They could start by, say, deleting all their files, then move on to sending profanity-laden email messages to their boss.

As I noted, COM class registrations are a commonly-reported vector for this attack, sometimes even touted as a way to obtain elevation. But that doesn’t work because COM is careful not to use registrations from `HKEY_ CURRENT_ USER` when running elevated. Only `HKEY_ LOCAL_ MACHINE` registrations are consulted when elevated, and attacking those registry key require that you already be elevated, so you haven’t gained anything.

Another place people report this type of false vulnerability is when they see that the `HKEY_ CURRENT_ USER` registry keys are affecting the behavior of `svchost.exe` processes. But you need to look more closely at *which* `svchost.exe` processes are affected.

Windows supports services that run under the context of the logged-on user, rather than as a privileged account. These reports breathlessly report that they found a way to inject code into `svchost.exe` via `HKEY_CURRENT_USER` attacks, but they failed to observe that the `svchost.exe` they attacked is running as the logged-on user. Again, all they did was attack their own process; there is no elevation of privilege.

Raymond Chen

Follow