It rather involved being on the other side of this airtight hatchway: Silently enabling features

devblogs.microsoft.com/oldnewthing/20121121-00

November 21, 2012

Raymond Chen

A security vulnerability report arrived which went roughly like this:

> When you programmatically enable the XYZ feature, the user receives no visual alert that it is enabled. As a result, malware can enable this feature and use it as part of an attempt to turn the machine into a botnet zombie. The XYZ feature should notify the user when it is enabled, so that to presence of malware is more easily determined.

Okay, first of all, before we get to the security part of this issue, let's look at the user interface design. The proposed change is that, when the XYZ feature is enabled programmatically, the user receive a notification "XYZ is now enabled." You know what most users are going to do when they get that notification? Ignore it. There are two cases where XYZ can be programmatically enabled. The user may have enabled it themselves by, say, checking a checkbox, and the code that handles the checkbox turns around and programmatically enables the XYZ feature. In this case, the notification is an annoyance like my three-year-old niece who narrates every single thing she does. The user goes to the XYZ control panel, enables XYZ, and in response to the XYZ control panel enabling XYZ, the user gets a notification balloon that says "XYZ is now enabled." Well DUH. The other case is that the user did *not* enable it themselves, in which case the balloon is an annoyance because it says something that the user doesn't care about and probably doesn't even understand. "The tech tech tech is now tech tech tech." Displaying a notification doesn't really help. Either the user expects it, in which case it's an annoyance, or the user doesn't expect it, in which case they most likely won't understand it either, so it's still just an annoyance. (And taking no action leaves the feature enabled.) Okay, now let's look at the security aspect of this report. Enabling the XYZ feature requires administrator privileges, so any malware which successfully turns on the XYZ feature has already pwned your machine. *It's already on the other side of the airtight hatchway*. Displaying a warning when your machine is pwned doesn't accomplish anything: Since the malware already has complete control of the machine, it can patch out the code that displays the notification balloon. In other words, the only case in which the user actually sees the XYZ notification is when the user was expecting it to be turned on anyway, at which point you're just being a chatty Cathy.

Exercise: “You can get rid of the notification in the case where the user enabled the feature manually adding a `fSuppressWarnings` parameter to the `EnableXYZ` function, and have the code that handles the checkbox pass `fSuppressWarnings = TRUE`. That leaves only the second case, which is exactly the case we want the user to be annoyed.” Discuss.

Raymond Chen

Follow