Solutions that don't actually solve anything

devblogs.microsoft.com/oldnewthing/20060510-06

May 10, 2006

Raymond Chen

If changing a setting requires administrator privileges in the first place, then any behavior that results cannot be considered a security hole because in order to alter the setting, attackers must already have gained administrative privileges on the machine, at which point you've already lost the game. If attackers have administrative privileges, they're not going to waste his time fiddling with some setting and leveraging it to gain even more privileges on the system. They're **already the administrator**; why go to more work to get what they already have? One reaction to this is to try to "secure" the feature by asking, "Well, can we make it harder to change that setting?" For example, in response to the Image File Execution Options key, Norman Diamond suggested "only allowing the launching of known debuggers." But this solution doesn't actually solve anything. What would a "known debugger" be?

- "The operating system contain a hard-coded list of known debuggers. On that list are ntsd.exe, cdb.exe, and maybe windbg.exe." Personally, I would be okay with that, but that's because I do all my debugging in assembly language anyway. Most developers would want to use devenv.exe or bds.exe or even gdb.exe. If somebody comes up with a new debugger, they would have to petition Microsoft to add it to the hard-coded list of "known debuggers" and then wait for the next service pack for it to get broad distribution. And even before the ink was dry on the electrons, I'm sure somebody somewhere will already have filed an anti-competitive-behavior lawsuit. ("Microsoft is unlawfully raising the barrier to entry to competing debugging products!")
- "Okay, then the program just needs to be digitally signed in order to be considered a 'known debugger'." Some people would balk at the $500/year cost of a code signing certificate. And should the operating system ask the user whether or not they trust the signing authority before running the debugger? (What if the debugger is being invoked on a service or a remote computer? There is nowhere to display the UI!) Actually, these were all trick questions. It doesn't matter whether the operating system prompts or not, because the attackers would just mark their signing certificate as a "trusted" certificate. And in fact the $500/year wouldn't stop the attackers, since they would just create their own certificate and install it as a "trusted root". Congratulations, the only people who have to pay the $500/year are the honest ones. The bad guys just slip past with their self-signed trusted-root certificate.

- “Okay, forget the digital signature thing, just have a registry key that lists all the ‘known debuggers’. If you’re on the list, then you can be used in Image File Execution Options.” Well, in that case, the attackers would just update the registry key directly and set themselves as a “known debugger”. That “known debuggers” registry key didn’t slow them done one second.
- “Okay, then not a registry key, but some other setting that’s hard to find.” Oh, now you’re advocating security through obscurity?

Besides, it doesn’t matter how much you do to make the Image File Execution Options key resistant to unwanted tampering. If the attacker has administrative privileges on your machine, they won’t bother with Image File Execution Options anyway. They’ll just install a rootkit and celebrate the addition of another machine to their robot army.

Thus is the futility of trying to stop someone who already has obtained administrative privileges. You’re just closing the barn door after the horse has bolted.

Raymond Chen

Follow