How about not granting debug privileges to users?

devblogs.microsoft.com/oldnewthing/20100104-00

January 4, 2010

Raymond Chen

Commenter Yuhong Bao suggests, "How about not granting debug privileges on the user? This will make bypassing the protection impossible." This is such a great idea that Windows has worked that way for years. Normal non-administrative users do not have debug privilege. They can only debug processes that they already have `PROCESS_ALL_ACCESS` to. In other words, non-administrative users can only pwn processes that they already pwn. No protection is being bypassed since you had full access in the first place. The SeDebugPrivilege allows you to debug any process, even those to which you do not have full access. This is clearly dangerous, which is why it's not granted to non-administrative users by default. Yuhong Bao also suggests, "How about separating the dangerous activities from the non-dangerous activities, or better, only allowing approved programs to do the dangerous activities?" (Where dangerous activities are defined as things that modify the program behavior.) I'm assuming this is discussing limiting the capabilities of SeDebugPrivilege, since in the absence of SeDebugPrivilege, the scope of your abilities is limited to things you already had the ability to do anyway; debugging didn't add anything new. But even if you limited SeDebugPrivilege to nondestructive actions, you can still lose the farm. This imaginary *SeLimitedDebugPrivilege* would still let you read a target process's memory, which means you can do things like steal passwords and snoop on the activities of other users. The last suggestion is to "only allow approved programs to do the dangerous activities." Again, I'm assuming this is discussing limiting the capabilities of SeDebugPrivilege, because without SeDebugPrivilege there is no new danger. But even in that limited context, what is an "approved program"? Approved by whom? Must the program be digitally signed by Microsoft? I suspect people who write debuggers which compete with, say, Microsoft Visual Studio, would be upset if they had to submit their debugger to Microsoft for approval. And what are the requirements for receiving this approval? Does the debugger have to pass some battery of tests like WHQL? There are already plenty of readers of this Web site who reject WHQL as useless. Would this "debugger certification" also be useless? Or maybe approval consists of merely being digitally signed at all? There are plenty of readers of this Web site who object to the high cost of obtaining a digital certificate (US$399; I don't think the $99 discounted version works for code signing.) And there are also plenty of readers who consider code signing to be payola and designed to maximize profit rather than effectiveness.

Or do you mean that the program needs to be listed in some new registry key called something like `Approved Debuggers`? Then what's to stop a rogue program from just auto-approving itself by writing to the Approved Debuggers registry key on its own?

But then again, all this is a pointless discussion once you realize that SeDebugPrivilege is granted by default only to administrators. And since administrators already pwn the machine, there's no protection that SeDebugPrivilege bypasses: You already bypassed it when you became an administrator.

Raymond Chen

Follow