If you grant somebody SeDebugPrivilege, you gave away the farm

devblogs.microsoft.com/oldnewthing/20080314-00

March 14, 2008

Raymond Chen

By default, users can debug only processes that they own. In order to debug processes owned by other users, you have to possess the SeDebugPrivilege privilege. But don't grant this privilege casually, because once you do, you gave away the farm. If you let users debug processes owned by other users, then they can debug processes owned by System, at which point they can inject code into the process and perform the logical equivalent of `net localgroup administrators anybody /add`, thereby elevating themselves (or anybody else) to administrator. If SeDebugPrivilege is equivalent to granting administrator privileges, why does it exist at all? It's not so much to protect the system as it is to protect the user. It's like the safety cover over the emergency power-off button. The purpose is merely to prevent you from pushing the button by mistake. If you're a developer debugging a service, you can turn on SeDebugPrivilege so you can debug the service, without turning on all the other administrative privileges. That way, you can't accidentally delete a critical file, for example.

The security investigations team have to deal with people who fail to understand that SeDebugPrivilege is (from a security perspective) equivalent to administrator. In one case, the finder contacted not the Microsoft Security Response Center but rather sent their "exploit" to high-level executives, who then had to transfer the issue to MSRC. And when informed of the misunderstanding, the finder just responded with a one-page rambling manifesto. As much as you are tempted to "just ignore the crazy guy", every reported security vulnerability must still be taken seriously. There might be something in there buried inside all the craziness. After all, they said Galileo was crazy.

Raymond Chen

Follow