Of what possible legitimate use are functions like CreateRemoteThread, WriteProcessMemory, and VirtualProtectEx?

devblogs.microsoft.com/oldnewthing/20120808-00

August 8, 2012

Raymond Chen

There are a bunch of functions that allow you to manipulate the address space of other processes, like `WriteProcessMemory` and `VirtualAllocEx`. Of what possible legitimate use could they be? Why would one process need to go digging around inside the address space of another process, unless it was up to no good? These functions exist for debuggers. For example, when you ask the debugger to inspect the memory of the process being debugged, it uses `ReadProcessMemory` to do it. Similarly, when you ask the debugger to update the value of a variable in your process, it uses `WriteProcessMemory` to do it. And when you ask the debugger to set a breakpoint, it uses the `VirtualProtectEx` function to change your code pages from read-execute to read-write-execute so that it can patch an `int 3` into your program. If you ask the debugger to break into a process, it can use the `CreateRemoteThread` function to inject a thread into the process that immediately calls `DebugBreak`. (The `DebugBreakProcess` was subsequently added to make this simpler.) But for general-purpose programming, these functions don’t really have much valid use. They tend to be used for nefarious purposes like DLL injection and cheating at video games.

[Raymond is currently away; this message was pre-recorded.]

Raymond Chen

Follow