Why is there a 64KB no-man's land near the end of the user-mode address space?

devblogs.microsoft.com/oldnewthing/20141009-00

October 9, 2014

Raymond Chen

We learned some time ago that there is a 64KB no-man's land near the 2GB boundary to accommodate a quirk of the Alpha AXP processor architecture. But that's not the only reason why it's there. The no-man's land near the 2GB boundary is useful even on x86 processors because it simplifies parameter validation at the boundary between user mode and kernel mode by taking out a special case. If the 64KB zone did not exist, then somebody could pass a buffer that straddles the 2GB boundary, and the kernel mode validation layer would have to detect that unusual condition and reject the buffer. By having a guaranteed invalid region, the kernel mode buffer validation code can simply validate that the starting address is below the 2GB boundary, then walk through the buffer checking each page. If somebody tries to straddle the boundary, the validation code will hit the permanently-invalid region and fail. Yes, this sounds like a micro-optimization, but I suspect this was not so much for optimization purposes as it was to remove weird boundary conditions, because weird boundary conditions are where the bugs tend to be.

(Obviously, the no-man's land moves if you set the /3GB switch.)

Raymond Chen

Follow