Throwing garbage on the sidewalk: The sad history of the rundll32 program

devblogs.microsoft.com/oldnewthing/20110909-00

September 9, 2011

Raymond Chen

During the development of Windows Vista, the application comaptibility team traced a bunch of issues back to people corrupting the stack by using the `rundll32` program to call functions that were not designed to be called by `rundll32`.

The problems were often subtle. For example, a batch file which used `rundll32` incorrectly ended up hanging because the `rundll32` process never returned. The misaligned stack resulted in registers being restored from the stack incorrectly, and then the cleanup code inside `rundll32` ends up getting confused and wedging itself. The programs got away with it on previous versions of Windows by sheer luck. The version of the compiler used by Windows Vista contains different optimizations, and it ended up arranging stack variables and using registers differently, and what in previous versions of Windows was some corruption that went largely unnoticed became corruption that resulted in the program getting stuck in an infinite loop. Lucky no longer.

I was asked to come up with a solution for this problem, to fix the `rundll32` program so it was more resilient to people who used it incorrectly. To *fix other people's bugs for them*.

The solution: Before calling the function, push a hundred bytes of garbage onto the stack (in case the called function pops too many bytes off the stack) and save the stack pointer in a global variable. After the function returns, restore the stack pointer, in case the called function pops too many or too few bytes off the stack. I think I may even have saved the processor registers in global variables, I forget.

Do not consider this free license to continue abusing the `rundll32` program. When the pet store opens on Sundays, that doesn't mean that it's okay to keep throwing garbage on the sidewalk.

Raymond Chen

Follow