Does ASLR relocate all DLLs by the same offset?

devblogs.microsoft.com/oldnewthing/20170118-00

January 18, 2017

Raymond Chen

I've seen multiple claims that the Windows implementation of ASLR chooses a single random offset and applies that same offset to all DLL base addresses.

> When the operating system loads, it applies a fixed random value to the DLL base. … The ASLR doesn't move DLL randomly. Without ASLR, if you get collisions, then you will get them with ASLR.

> If two DLLs have base addresses to designed to place them consecutively, they'll still be consecutive even with ASLR.

In other words, the claim is that if you have two DLLs, call them DLL1 with base address *base1* and DLL2 with base address *base2*, then, assuming there are no base address collisions with already-loaded DLLs, ASLR will load the two DLLs at *base1* + *N* and *base2* + *N* for some value of *N* (possibly negative). In particular, this means that if *base1* and *base2* are adjacent, then the two DLLs will remain adjacent after ASLR, and if the two DLLs have colliding base addresses, then they will also have colliding base addresses after ASLR.

But it's not true, and as far as I can tell, it has never been true.

ASLR chooses the base address pseudo-randomly, though it does take some of the original base addresses into account. For example, if the original base address was below the 4GB boundary, then the new pseudo-random base address will also be below the 4GB boundary.

But it doesn't try to preserve relative base addresses. Each DLL is assigned a new pseudo-random base address independently. There is no correlation, or at least there is no conscious effort to correlate them.

Raymond Chen

Follow