User Address Space Layout

Just as address space in the kernel is dynamic, the user address space in Windows Vista and later versions is also built dynamically—the addresses of the thread stacks, process heaps, and loaded images (such as DLLs and an application's executable) are dynamically computed (if the application and its images support it) through a mechanism known as Address Space Layout Randomization, or ASLR.

At the operating system level, user address space is divided into a few well-defined regions of memory, shown in Figure 9-15. The executable and DLLs themselves are present as memory mapped image files, followed by the heap(s) of the process and the stack(s) of its thread(s). Apart from these regions (and some reserved system structures such as the TEBs and PEB), all other memory allocations are run-time dependent and generated. ASLR is involved with the location of all these regions and, combined with DEP, provides a mechanism for making remote exploitation of a system through memory manipulation harder to achieve—by having code and data at dynamic locations, an attacker cannot typically hardcode a meaningful offset.

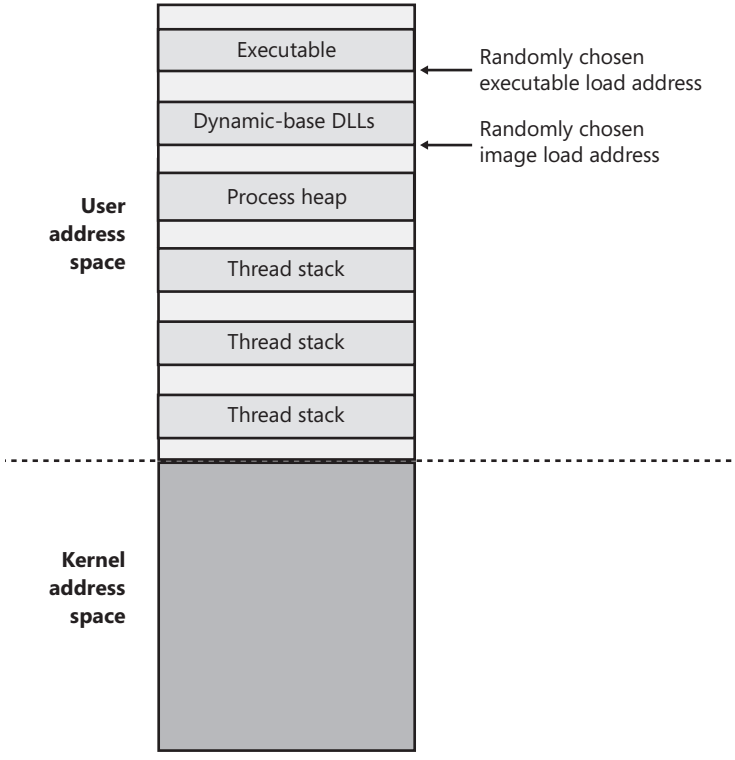

FIGURE 9-15 User address space layout with ASLR enabled