**What is do\_brk()?**

do\_brk()是一个内部的内核函数，用来管理进程的memory heap的增长，当heap过大时缩小它，就是为了防止heap无限增长以至于扩展到别的地方，覆盖别的东西。

它本来是用来优化do\_mmap()，因为没有边缘检测，使得用户可以把自己的堆扩展到内核空间，就成了一个可以exploit的bug。

The do\_brk() is an internal kernel function which is called indirectly to manage process' s memory heap (brk) growing or shrinking it accordingly.

The user may manipulate his heap with the brk(2) system call which calls do\_brk() internally.

The do\_brk() code is a simplified version of the mmap(2) system call and only handles anonymous mappings for uninitialized data.

**How to exploit?**

Step 1: Change Program Layout and Expand Heap over kernel.

先不停的扩展heap，让heap覆盖到内核空间。

Step 2: Find the memory we want to change. Create a new kpage with LDT\_mod technique. Scan memory using verr and signals technique.

找到要改的内核页，重写ldt

Step 3: Expand with do\_brk() to page table and turn off s-bit on that page.

Step 4: Setup call gate which enables privilege level transition from the user to the kernel privilege level.

Step 5: Trampoline Code

Step 6: Scan task\_struct to set euid, etc to 0

Step 7: Cleanup

Step 8: Shell, Rooted