CS392 – Quiz

Name: **Maheeth Reddy**

Roll No.: **1801CS31**

Date: **18-Apr-2021**

Ans 1:

The given program attack.c, consists of a mistake. We are initializing the map variable (void pointer) in the main() function using mmap() function. One of the parameters to mmap() is mentioned as MAP_SHARED flag. This causes the mapped memory to be shared between two processes. But, this flag is supposed to be MAP_PRIVATE for the attack to happen successfully.

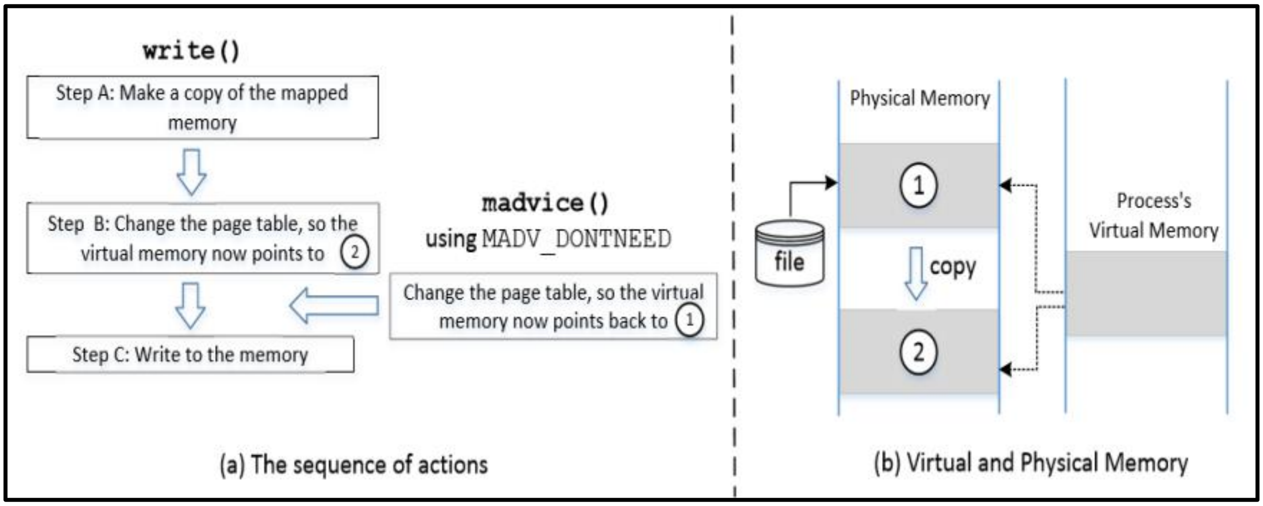
Shankar is trying to do a Dirty-COW attack. The three important steps in this attack are:

(A) Make a copy of the mapped memory

(B) Update the page table, so the virtual memory points to newly created physical memory

(C) Write to the memory.

The principle behind the attack is to create a race condition like TOCTTOU. The steps A,B,C are not atomic in nature: they can be interrupted by other threads which creates a potential race condition leading to Dirty Cow vulnerability.1


write()
Step A: Make a copy of the mapped memory
Step B: Change the page table, so the virtual memory now points to 2
Step C: Write to the memory
madvice()
using MADV_DONTNEED
Change the page table, so the virtual memory now points back to 1
(a) The sequence of actions
Physical Memory
1
copy
2
file
Process's Virtual Memory
(b) Virtual and Physical Memory

Picture taken from slides

Consider the following scenario. If madvise() is executed between Steps B and C, Step B will make the virtual memory point to 2. But madvise() will change it back to 1, sort of negating Step B. This causes Step C to modify the physical memory marked by 1, instead of the private copy. Now, the read-only file is modified, which Shankar had no permission to edit. Since this is a COW memory, when the write() system call is invoked, it triggers A,B,C without a double check.2