Dirty COW

S8.1. A file's content is a string "`Hello World`". When this file is mapped to memory (the entire file) using `mmap()`, and the memory address is stored in a variable `map`. Please describe what the following `printf()` statement prints out.

```
char *addr = (char *)map;
printf("%s\n", map +6);
```

S8.2. The `fork()` system call creates a new process from a parent process. The new process, i.e., the child process, will have a copy of the parent process's memory. Typically, the memory copy is not performed when the child process is created. Instead, it is delayed. Please explain when the memory copy will occur.

S8.3. When a process maps a file into memory using the `MAP_PRIVATE` mode, the memory mapping is depicted in Figure 1. (1) Please describe what is going to happen when this process writes data to address `0x5100`. (2) The Dirty COW race condition occurs inside the `write()` system call. Please explain exactly where the problem is. (3) How can this race condition vulnerability be exploited?

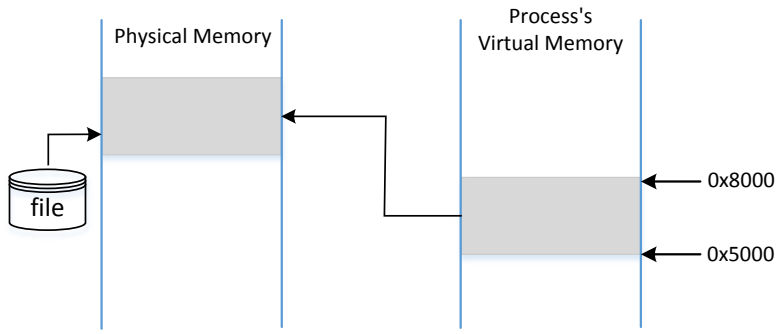

Figure 1: Figure for Problem S8.3.

S8.4. The permission of the file `/home/seed/zzz` is readable and writable to the user `seed`. Does the following code (executed by `seed`) modify the content of `/home/seed/zzz`?

```
int f=open("/home/seed/zzz", O_RDWR);
fstat(f, &st);
// Map the entire file to memory
map=mmap(NULL, st.st_size, PROT_READ|PROT_WRITE,
                           MAP_PRIVATE, f, 0);
memcpy(map, "new content", strlen("new content"));
```

S8.5. In the Dirty COW attack, can we run two processes, instead of two threads?

S8.6. In this chapter, we show that by exploiting the Dirty COW race condition, we can modify the `/etc/passwd` file and gain the root privilege. Please name two other files that can be attacked to gain the root privilege.

S8.7. If we use the `MAP_PRIVATE` to map a read-only file to the memory, and then use `memcpy()` to write to it. Will this cause copy-on-write?

S8.8. ★★
Why cannot we implement copy-on-write in `memcpy()`, so we can use it to write to a private copy of the mapped memory?