

CS5460 – bonus Assignment



April 17, 2016

Varun GATTu

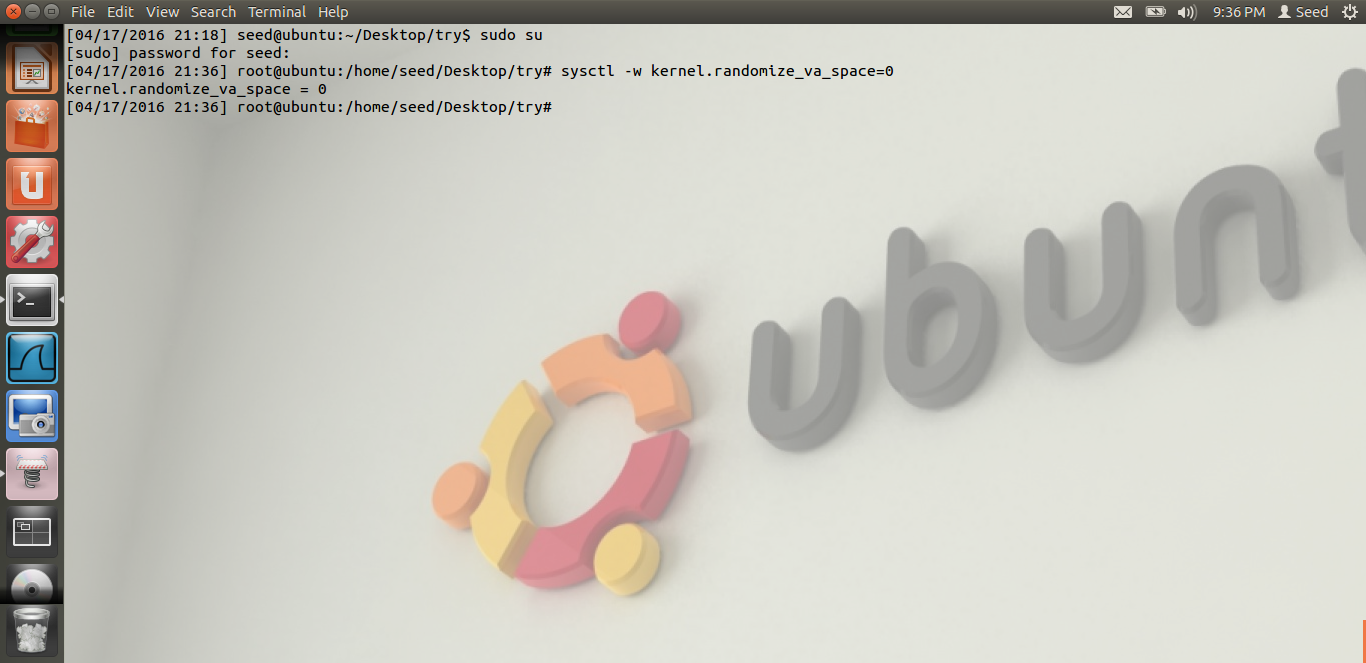
A02092613

**Cross site scripting (XSS) attack Lab**

**2. Lab Tasks**

**2.1. Lab Environment**

Before we start the return to libc attack, similar to buffer overflow lab; we initially disable the address randomization feature of Ubuntu with the command shown in the following figure:



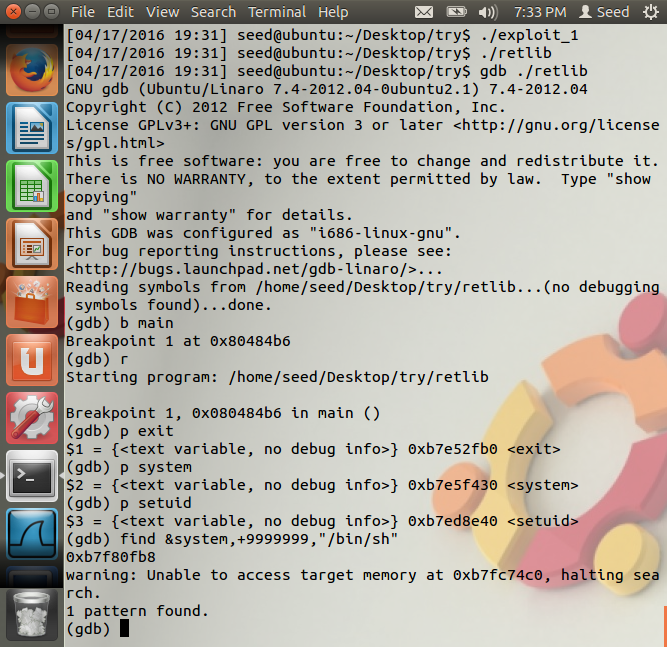
**2.2. The Vulnerable Program**

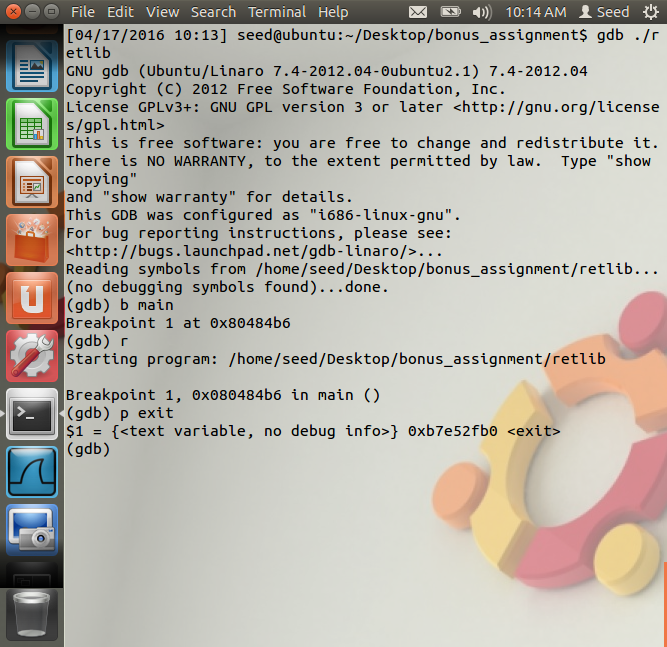
We then write a program which has buffer overflow vulnerability and try to exploit it. We create a buffer of size 12 and try to enter 40 characters into it. The program is compiled as shown:

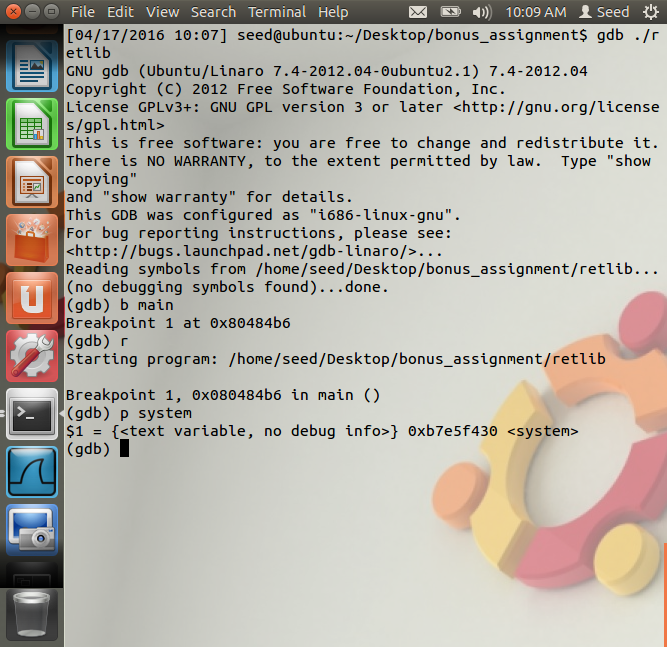


**2.3. Task 1: Exploiting the Vulnerability**

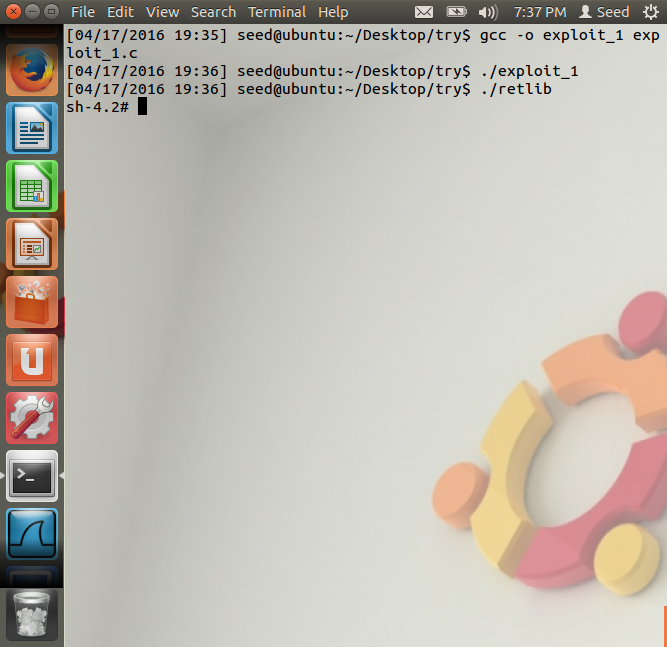
Here, we try to exploit the vulnerability that we have from retlib program. We have tried to get the stack address using the overflow program. We find the addresses of system(), exit() and /bin/sh and input it to the exploit program. They are found using gdb compiler which is explained in the later part of the assignment.



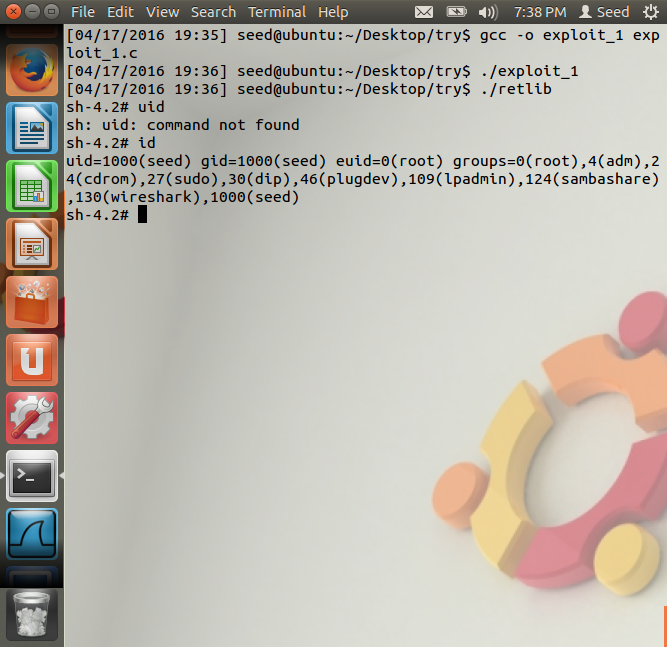




Now, we compile the exploit program and then run both the exploit and the vulnerable program as follows:



With the id command, we display the uid of the user as follows:



**2.4. Task 4: Protection in /bin/bash**

Here, we make the /bin/sh point to /bin/bash and check if we will be able to get the root shell. Then we make small changes to the exploit program and add the uid to be zero. Then, we recompile and run the program. This will not lead to the root shell because bash has some feature which protects the stack.



**2.5. Task 3: Address Randomization and Stack smash protection**

Here, we enable the address randomization which was disabled in the 1st task and try to recompile and run the program. This time, we observe that we cannot achieve the root shell as shown in this image:

