Lab1 BufferOverflow

Task 1: We provide you with a completed exploit code called “exploit.py”. You need to adjust the variables of the program accordingly and fill in any missing code to fulfill the buffer overflow attack.

Notes on python, you don't need to compile a .py file to run it. Python is an interpreted language, and you can run the scripts directly, either using:
python exploit.py

Or make your script executable by adding *"#!/usr/bin/python3"* to the top of the script, making the file executable with "*chmod u+x exploit.py*" and then running:
./exploit.py

The book utilizes the second version ./exploit.py

After you finish adjusting the above program, run it. This will generate the contents for “badfile”. Then compile and run the vulnerable program "stack.c". If your exploit is implemented correctly, you should be able to get a shell:

$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed)

Task2: Find a way to obtain root shell (read book/slides).
you should get the following output:
VM# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed)

Task3: On 32-bit Linux machines, stacks only have 19 bits of entropy, which means the stack base address can have 2^19 = 524, 288 possibilities. This number is not that high and can be exhausted easily with the brute-force approach. In this task, we use such an approach to defeat the address randomization countermeasure on our 32-bit VM. First, we turn on the Ubuntu’s address randomization using the following command:

sudo /sbin/sysctl -w kernel.randomize_va_space=2

Then use the shell script in the book to figure out how to attack the vulnerable program repeatedly.

To summarize, the lab goal is:
Task 1- edit the given source code “exploit.py” so that the buffer overflow attack is successful.
Task 2- ensure the shell is running as root.
Task 3- Defeat the Address Randomization applied by the operating system.

In your paper show and explain the following:
-The adjusted code segments, and describe what changes you made and why.
-How you obtained the needed addresses, show screenshots.
-The screenshots of the successful buffer overflow attack.
-How you made the shell run as root.
-How you defeated Address Randomization, screenshot.