CSED211, Fall 2021
Bomb Lab: Defusing a Binary Bomb
Assigned: Sept. 30, Due: Oct. 14 11:59PM

Jeongwoo Kim (`jwkim0417@postech.ac.kr`) is the lead person for this lab.

1 Introduction

The nefarious *Dr. Evil* has planted a slew of "binary bombs" on our class machines. A binary bomb is a program that consists of a sequence of phases. Each phase expects you to type a particular string on `stdin`. If you type the correct string, then the phase is *defused* and the bomb proceeds to the next phase. Otherwise, the bomb *explodes* by printing `"BOOM!!!"` and then terminating. The bomb is defused when every phase has been defused.

There are too many bombs for us to deal with, so we are giving each student a bomb to defuse. Your mission, which you have no choice but to accept, is to defuse your bomb before the due date. Good luck, and welcome to the bomb squad!

Step 1: Get Your Bomb

You can obtain your bomb from the following Onedrive:

```
https://bit.ly/3AOW7V0
```

In the drive, you will find *bombk.tar* for some integer k, where k is the remainder of dividing your student ID by 2222 (e.g. 20202616 % 2222 → 192, 20212927 % 2222 → 1615).

Save the *bombk.tar* file to a (protected) directory in which you plan to do your work (i.e. programming server). Then give the command: `tar -xvf bombk.tar`. This will create a directory called `./bombk` with the following files:

- `README`: Identifies the bomb and its owners.
- `bomb`: The executable binary bomb.
- `bomb.c`: Source file with the bomb's main routine and a friendly greeting from Dr. Evil.

After getting your bomb, please check the README and ID file if it is correct.

Step 2: Defuse Your Bomb

Your job for this lab is to defuse your bomb.

You must do the assignment on your own programming server. You can use many tools to help you defuse your bomb. The best way is to use your favorite debugger (e.g. gdb) to step through the disassembled binary.

The first four phases are worth 10 points each. Phases 5 and 6 are a little more difficult, so they are worth 15 points each. So the maximum score you can get is 70 points.

Although phases get progressively harder to defuse, the expertise you gain as you move from phase to phase should offset this difficulty. However, the last phase will challenge even the best students, so please don't wait until the last minute to start.

The bomb ignores blank input lines. If you run your bomb with a command line argument, for example,

```
linux>  ./bomb solution.txt
```

then it will read the input lines from *solution.txt* until it reaches EOF (end of file), and then switch over to *stdin*. In a moment of weakness, Dr. Evil added this feature so you don't have to keep retyping the solutions to phases you have already defused.

To avoid accidentally detonating the bomb, you will need to learn how to single-step through the assembly code and how to set breakpoints. You will also need to learn how to inspect both the registers and the memory states. One of the nice side-effects of doing the lab is that you will get very good at using a debugger. This is a crucial skill that will pay big dividends the rest of your career.

Handin

This is an individual project. All handins are electronic. By the due date, you need to hand in your assignment to PLMS.

- `compressed file`: The contents should ideally be such that we can defuse the entire bomb simply running `./bomb solution.txt`, or such that it will pass some number of phases and then blow up. Compress the *solution.txt* file with your original bomb file (ID, bomb, and bomb.c) and hand in the compressed file. The format of file is (student number)_(your name).zip.
- `report`: Explain *in detail* how each phase is defused. The format of file is (student number)_(your name).doc / .pdf