CSED 211 Fall 2019

Lab Assignment #3: Defusing a Binary Bomb

Assigned: September 18, Due: October 1 pm 23:59

Introduction

The nefarious Joker, the Clown Prince of Crime, 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

https://drive.google.com/open?id=1zEYZ9pYC8o_w6yzjpk1KnAU5wHGFjbAG

In our link, you will find ***bombk.tar*** for some integer ***k***. ***k*** is your attendance number. You can know

your attendance number in LMS → 수강생조회.

Save the ***bombk.tar*** file to a Linux directory in which you plan to do your work (maybe, *programming.postech.ac.kr*.) Then give the command: ***tar –xvf bombk.tar.*** This will create a directory called ***./bombk*** with the following files:

- ID: Your POVIS ID
- bomb: The executable binary bomb.
- bomb.c: Source file with the bomb's main routine and a friendly greeting from the Joker

After getting your bomb, please check the README and ID file in order to whether the bomb is yours or not.

Step 2: Defuse Your Bomb

Your job for this lab is to defuse your bomb.

The best way is to use your favorite debugger 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 from the defuse steps is 70 points. (Actually, there is a secret phase. If you solve this phase, then you will get bonus 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 psol.txt
```

then it will read the input lines from ***psol.txt*** until it reaches EOF (end of file), and then switch over to ***stdin***. In a moment of weakness, the Joker 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

Hand in

This is an individual project. By the due date, you need to hand in your assignment to LMS.

- A file, ***solution.txt***. The contents should ideally be such that we can defuse the entire bomb simply by running ***./bomb solution.txt,*** or such that it will pass some number of phases and then blow up.

  Please compress ***solution.txt*** file with your original bomb file (ID, bomb, and bomb.c) and hand in the compressed file.

- **Final report**. There is no report format, however, please explain **in detail** how each phase is defused.