CSC3050 Fall 2024
Project2: Defusing a Binary Bomb
Assigned: Oct. 8th, Due: 23:59 Monday Oct. 21

Many thanks to previous USTFs Junlin Huang and Vincent Janssen for the initial version of this project.

USTFs Letian Cheng (121090088@link.cuhk.edu.cn) and Han Yan (122090897@link.cuhk.edu.cn) are responsible for this project.

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 by pointing your Web browser at:

```
http://proj2.eastasia.cloudapp.azure.com/
```

This will display a binary bomb request form for you to fill in. Enter your user name and email address and hit the Submit button. The server will build your bomb and return it to your browser in a `tar` file called `bombk.tar`, where k is the unique number of your bomb.

The bomb will be run in the provided QEMU environment in the form of a Docker container. To set it up, simply pull the container image from `ghcr.io/janssen-v/rvemu` and follow the instructions on the repository: `https://github.com/janssen-v/rvemu` to setup your environment.

Figure 1: This is request format example.

Save the `bombk.tar` file to a (protected) directory in which you plan to do your work. 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. (You can think of it as the .exe file)
- `bomb.c`: Source file with the bomb's main routine and a friendly greeting from Dr. Evil.

The source code of everything else will be hidden from you. You would have to defuse your bomb based on the executable file only. Before starting the lab, we recommend you to generate a full disassembly of the bomb using the command `objdump -d bomb > bomb.asm`. This will likely result in an assembly file of 2000+ lines. However, do not be taken aback because **most of the code are useless** to your debugging process! It will be up to you to find useful threads of clues amidst the nightmarish quagmire of tangled logic.

If for some reason you request multiple bombs, this is not a problem. Choose one bomb to work on and delete the rest. (but please **do not order 10 bombs** or sth, which **might affect your score.**) Do note that every single bomb is unique. No matter how many bombs you require, every single one has completely different answers.

Step 2: Defuse Your Bomb

Your job for this lab is to defuse your bomb.

You must run your bomb with a connection to our server, so make sure you have internet connection while you are defusing your bomb. In fact, there is a rumor that Dr. Lethal is so lethal that the bomb will always blow up if run elsewhere. If you submit a solution on BB without the defusal being logged on our server, we will not consider it as a valid submission. There are several other tamper-proofing devices built into the bomb as well, or so we hear.

You can use many tools to help you defuse your bomb. Please look at the **hints** section for some tips and ideas. The best way is to use your favorite debugger to step through the disassembled binary.

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.

Grading Criteria

The maximum score you can get for this lab is 100 points.The first four phases are worth 15 points each. **Phases 5 and 6 are a little more difficult**, so they are worth 20 points each. As long as you have came to our tutorials and understood the practice bomb, you should at least be able to do the first four phases.

Each time your bomb explodes, it will notify the bomblab server, and you lose 1 point (up to a max of 20 points) in your final score. So there are consequences to exploding the bomb. In a moment of conscience, Dr. Lethal decided to give you 3 (three) free explosions. So each of you have three chances, and three chances only, to detonate the bomb without losing any points. Nevertheless, please be careful!

The deadline for this lab is **23:59, Monday, 2024/10/21**. For each day after the deadline, 10 points will be deducted from your final score up to 30 points, after which you will get 0 point.

Submission

This is an individual project. All handins are electronic. The bomb will notify the CSC3050 teaching team automatically about your progress as you work on it. You can keep track of how you are doing by looking at the class scoreboard at:

`http://proj2.eastasia.cloudapp.azure.com/scoreboard`

This web page is updated continuously to show the progress for each bomb. For submission to BB, you would need to create a solution.txt, which contains the answers for each phase on each line and nothing else (**refer to the one in the practice bomb**). Put the solution.txt file in your bombk directory, zip it up using the command

```
tar -cvf bombk.tar bombk
```

and submit the whole thing. Your submission should have the following file structure:

```
bombk.tar/
    bombk/
```

```
|-
|---bomb
|---bomb.c
|---README
|---solution.txt // your solution to the bomb
|-
```