15-213 Fall 20xx
15-213 秋季 20xx
Lab Assignment L2: Defusing a Binary Bomb
实验任务 L2: 拆除二元炸弹
Assigned: Sept. 13, Due: Friday Sept. 22
分配时间: 9 月 13 日截止日期: 9 月 22 日星期五

Harry Bovik (`bovik@cs.cmu.edu`) is the lead person for this lab.
Harry Bovik (Bovik@cs.cmu.edu)是这个实验室的负责人。

1 Introduction
1 引言

The nefarious *Dr. Evil* has planted a slew of "binary bombs" on our class machines. A b inary 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.
邪恶的邪恶博士在我们班的机器上安放了大量的"二元炸弹"。二进制炸弹是一个由一系列阶段组成的程序。每个阶段都要求你在 stdin 上键入一个特定的字符串。如果你键入了正确的字符串，那么这个阶段就会被拆除，炸弹就会进入下一个阶段。否则，炸弹会通过打印" BOOM 然后终止。当每个阶段都被拆除时，炸弹就会被拆除。

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!
我们要处理的炸弹太多了，所以我们给每个学生一个炸弹去拆除。你们的任务，你们别无选择，只能接受，就是在预产期前拆除炸弹。祝你好运，欢迎加入拆弹小组！

Step 1: Get Your Bomb
第一步: 找到你的炸弹

You can obtain your bomb by pointing your Web browser at:
您可以通过将浏览器指向以下方面来获取炸弹:

```
http://$Bomblab::SERVER_NAME:$Bomblab::REQUESTD_PORT/
Http://$bomblab: : server _ name: $bomblab: : requestd _ port/
```

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.
这将显示一个二进制炸弹请求表单供您填写。输入你的用户名和电子邮件地址，然后点击提交按钮。服务器将建立你的炸弹，并返回到你的浏览器在一个叫做 bomk.tar 的 tar 文件，其中 k 是你的炸弹的唯一编号。

Save the `bombk.tar` file to a (protected) directory in which you plan to do your wor k. Then give the command: `tar -xvf bombk.tar`. This will create a directory called `./bombk` with the following files:
将 bomk.tar 文件保存到一个(受保护的)目录中，你计划在这个目录中完成你的工作 k。然后给出以下命令: tar-xvf bomk.tar。这将创建一个名为。/bomk，包含以下文件:

- `README`: Identifies the bomb and its owners.
  自述文件： 识别炸弹和它的主人。

- `bomb`: The executable binary bomb.
  炸弹： 可执行的二进制炸弹。

• `bomb.c`: Source file with the bomb's main routine and a friendly greet ing from Dr. Evil.
炸弹 `c`：源文件，炸弹的主要程序和邪恶博士的友好问候。

• `writeup.{pdf,ps}`: The lab writeup.
注释。`{ pdf，ps }` ：实验室注释。

If for some reason you request multiple bombs, this is not a problem. Choose one bomb to work on and delete the rest.
如果出于某种原因，您要求多个炸弹，这不是一个问题。选择一个炸弹，然后删除剩下的。

Step 2: Defuse Your Bomb
第二步: 拆除炸弹

Your job for this lab is to defuse your bomb.
你在这个实验室的工作就是拆除炸弹。

You must do the assignment on one of the class machines. In fact, there is a rumor that Dr. Evil really is evil, and the bomb will always blow up if run elsewhere. There are several other tamper-proofing devices built into the bomb as well, or so we hear.
你必须在其中一台机器上完成作业。事实上，有传言说邪恶博士真的很邪恶，如果把炸弹运到别处，它总是会爆炸。据我们所知，炸弹里还有其他一些防篡改装置。

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.
你可以使用许多工具来帮助你拆除炸弹。请看提示部分获取一些提示和想法。最好的方法是使用你最喜欢的调试器逐步通过反汇编的二进制文件。

Each time your bomb explodes it notifies the bomblab server, a nd you lose 1/2 point (up to a max of 20 points) in the final score for the lab. So there are consequenc es to exploding the bomb. You must be careful!
每次你的炸弹爆炸，它通知炸弹服务器，你失去了 1/2 点(最多 20 点)的最终分数为实验室。所以引爆炸弹是有后果的。你一定要小心！

The first four phases are worth 10 points each. Phases 5 and 6 ar e a little more difficult, so they are worth 15 points each. So the maximum score you can get is 70 points.
前四个阶段各值 10 分。第五阶段和第六阶段比较困难，所以每个阶段都值 15 分。所以你能得到的最高分是 70 分。

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 c hallenge even the best students, so please don't wait until the last minute to start.
虽然阶段变得越来越难以化解，但是你从一个阶段到另一个阶段所获得的专业知识应该可以抵消这个困难。然而，最后一个阶段甚至会挑战最优秀的学生，所以请不要等到最后一分钟才开始。

The bomb ignores blank input lines. If you run your bomb with a command line argument, for example,
炸弹会忽略空白输入行。例如，如果您使用命令行参数运行炸弹，

```
linux>  ./bomb psol.txt
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, Dr. Evil added this feature so you don't have to keep retyping the solutions to phases you have already defused.
然后它将从 psol.txt 读取输入行，直到达到 EOF (文件结束)，然后切换到 stdin。在一个软弱的时刻，邪恶博士添加了这个功能，这样你就不必一直重复输入你已经拆除的阶段的解决方案。

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.
为了避免意外引爆炸弹，你需要学习如何单步通过汇编代码和如何设置断点。你还需要学习如何检查寄存器和内存状态。做实验的一个好的副作用就是你会非常擅长使用调试器。这是一项至关重要的技能，将在你的职业生涯中带来丰厚的回报。

Logistics
后勤

This is an individual project. All handins are electronic. Clarifications and corrections will be posted on the course message board.
这是一个单独的项目。所有的手都是电子的。澄清和更正将张贴在课程留言板上。

Handin
汉丁

There is no explicit handin. The bomb will notify your instructor 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:
没有明确的 handin。炸弹会自动通知你的教练你的进展，因为你的工作。你可以通过查看课堂记分板来记录你的表现:

```
http://$Bomblab::SERVER_NAME:$Bomblab::REQUESTD_PORT/scoreboard
   Http://$bomblab: : server _ name: $bomblab: : requestd _
                        port/scoreboard
```

This web page is updated continuously to show the progress for each bomb.
这个网页不断更新以显示每个炸弹的进度。

Hints *(Please read this!)*
提示(请读这个!)

There are many ways of defusing your bomb. You can examine it in great detail without ever running the program, and figure out exactly what it does. This is a useful t echnique, but it not always easy to do. You can also run it under a debugger, watch what it does step by step, and use this information to defuse it. This is probably the fastest way of defusing it.
拆除炸弹的方法有很多。你可以在不运行程序的情况下仔细检查它，然后弄清楚它到底是做什么的。这是一个有用的 t 技术，但并不总是很容易做到。你也可以在调试器下运行它，一步一步地观察它是怎么做的，然后利用这些信息来缓解它。这可能是最快的解决方法。

We do make one request, *please do not use brute force!* You could write a program that will try every possible key to find the right one. But this is no good for sever al reasons:
我们有一个请求，请不要使用蛮力！你可以写一个程序，尝试每一个可能的关键，以找到正确的。但是由于以下几个原因这样做是不好的:

- You lose 1/2 point (up to a max of 20 points) every time you guess incorrectly and the bomb explodes.
每当你猜错一次，炸弹就会爆炸，你就会损失 1/2 点(最多 20 点)。

- Every time you guess wrong, a message is sent to the bomblab server. You could very quickly saturate the network with these messages, and cause the system administrators to revoke your computer access.
每次你猜错了，就会有一条信息发送到炸弹服务器。你可以很快用这些信息使网络饱和，并导致系统管理员取消你的计算机访问权限。

- We haven't told you how long the strings are, nor have we told you what characters are in them. Even if you made the (incorrect) assumptions that they all are less than 80 characters long and only contain
我们没有告诉你这些字符串有多长，也没有告诉你里面有什么字符。即使你做了(不正确的)假设，它们都小于 80 个字符长，并且只包含

  letters, then you will have 26^{80} guesses for each phase. This will take a very long time to run, and you will not get the answer before the assignment is due.

字母，然后你将有 2680 个猜测，每个阶段。这将需要很长的时间来运行，你不会得到答案之前的作业。

There are many tools which are designed to help you figure out b oth how programs work, and what is wrong when they don't work. Here is a list of some of the tools you may find useful in analyzing your bomb, and hints on how to use them.
有很多工具可以帮助你弄清楚程序是如何工作的，以及程序不工作时出了什么问题。这里列出了一些你可能会发现在分析炸弹时有用的工具，以及如何使用它们的提示。

- `gdb`

`Gdb`

The GNU debugger, this is a command line debugger tool available on virtually every platform. You can trace through a program line by line, examine memory and registers, look at both the source code and assembly code (we are not giving you the source code for most of your bomb), set breakpoints, set memory watch points, and write scripts.
GNU 调试器，这是一个命令行调试工具，可用于几乎每个平台。您可以逐行跟踪程序，检查内存和寄存器，查看源代码和汇编代码(我们不会为您提供大部分炸弹的源代码)，设置断点，设置内存观察点，并编写脚本。

The CS:APP web site

`http://csapp.cs.cmu.edu/public/students.html`

政务司司长办公室: APP 网站

http://csapp.CS.cmu.edu/public/students. html

has a very handy single-page `gdb` summary that you can print out and use as a reference. Here are some other tips for using `gdb`.
有一个非常方便的单页 gdb 摘要，你可以打印出来作为参考。下面是一些使用 gdb 的其他提示。

– To keep the bomb from blowing up every time you type in a wrong input, you'll want to learn how to set breakpoints.

为了防止每次输入错误时炸弹爆炸，您需要学习如何设置断点。

– For online documentation, type " `help`" at the `gdb` command prompt, or type " `man gdb`", or " `info gdb`" at a Unix prompt. Some people also like to run `gdb` under `gdb-mode` in `emacs`.

- 对于在线文档，在 gdb 命令提示符处键入" help"，或在 Unix 提示符处键入" man gdb" 或" info gdb"。有些人也喜欢在 emacs 的 gdb 模式下运行 gdb。

• `objdump -t`
Objdump-t

This will print out the bomb's symbol table. The symbol table includes the names of all functions and global variables in the bomb, the names of all the functions the bomb calls, and their addresses. You may learn something by looking at the function names!
这将打印出炸弹的符号表。符号表包括炸弹中所有函数和全局变量的名称，炸弹调用的所有函数的名称，以及它们的地址。你可以通过查看函数名来学到一些东西！

• `objdump -d`
Objdump-d

Use this to disassemble all of the code in the bomb. You can also just look at individual functions. Reading the assembler code can tell you how the bomb works.
用这个代码来反汇编炸弹中的所有代码。你也可以看看单个的函数。阅读汇编代码可以告诉你炸弹是如何工作的。

Although `objdump -d` gives you a lot of information, it doesn't tell you the whole story. Calls to system-level functions are displayed in a cryptic form. For example, a call to `sscanf` might appear as:
虽然 objdump-d 提供了很多信息，但它并没有告诉你全部情况。对系统级函数的调用是以一种神秘的形式显示的。例如，对 sscanf 的调用可能会显示为:

```
8048c36:  e8 99 fc ff ff  call   80488d4 <_init+0x1a0>
8048c36: e899 fc ff ff ff 调用 80488d4 < _ init + 0 x1a0 >
```

To determine that the call was to `sscanf`, you would need to disassemble within `gdb`.
要确定调用是对 sscanf 的，需要在 gdb 中进行反汇编。

• `strings`
字符串

This utility will display the printable strings in your bomb.
此实用程序将显示炸弹中的可打印字符串。

Looking for a particular tool? How about documentation? Don't forget, the commands `apropos`, `man`, and `info` are your friends. In particular, `man ascii` might come in useful. `info gas` will give you more than you ever wanted to know about the GNU Assembler. Also, the web may also be a treasure trove of information. If you get stumped, feel free to ask your instructor for help.

寻找一个特定的工具？文档呢？别忘了，适当的命令，伙计，和信息是你的朋友。特别是，man ascii 可能会派上用场。Info gas 会给你更多你想知道的关于 GNU 汇编器的信息。此外，网络也可能是一个信息的宝库。如果你被难住了，请随时向你的导师寻求帮助。