# 15-213 Fall 20xx

Lab Assignment L2: Defusing a Binary Bomb Assigned: Sept. 13, Due: Friday Sept. 22

Harry Bovik (bovik@cs.cmu.edu) 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!

邪恶的邪恶博士在我们的班级机器上放置了大量的“二元炸弹”。二元炸弹是由一系列阶段组成的程序。每个阶段都要求您在stdin上键入一个特定的字符串。如果你输入了正确的字符串，那么这个阶段就被解除了，炸弹进入下一个阶段。否则，炸弹爆炸时会印上“轰！！！”然后终止。当每一个阶段都被拆除时，炸弹就被拆除了。

有太多的炸弹要我们处理，所以我们给每个学生一个炸弹来拆除。你的任务，你别无选择，只能接受，就是在到期日前拆除你的炸弹。祝你好运，欢迎加入拆弹小组！

## Step 1: Get Your Bomb

You can obtain your bomb by pointing your Web browser at:

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.

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:

您可以通过将Web浏览器指向以下位置获取炸弹：

http://$Bomblab::SERVER\_NAME:$Bomblab::REQUESTD\_PORT/

这将显示一个二进制炸弹请求表供您填写。输入您的用户名和电子邮件地址，然后单击“提交”按钮。服务器将构建你的炸弹，并将其返回到你的浏览器的tar文件名为重磅焦油，其中k是炸弹的唯一编号。

保存重磅焦油将文件保存到计划在其中执行工作的（受保护）目录。然后给出命令：tar-xvf重磅焦油. 这将创建一个名为./bombk的目录，其中包含以下文件：

* 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.炸弹的主要程序和邪恶博士友好问候的原始文件。
* writeup.{pdf,ps}: The lab writeup.实验报告。

If for some reason you request multiple bombs, this is not a problem. Choose one bomb to work on and delete the rest.

如果出于某种原因你要求多枚炸弹，这不是问题。选择一个炸弹工作，删除其余的。

## Step 2: Defuse Your Bomb

Your job for this lab is to defuse your bomb.

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.

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.

你在这个实验室的工作就是拆除你的炸弹。

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是邪恶的，如果在别处运行，炸弹总会爆炸。炸弹里还有其他几种防篡改装置，或者我们听说的。

你可以用很多工具来帮助你拆除炸弹。请看提示部分的一些提示和想法。最好的方法是使用您最喜欢的调试器来逐步检查反汇编的二进制文件。

Each time your bomb explodes it notifies the bomblab server, and you lose 1/2 point (up to a max of 20 points) in the final score for the lab. So there are consequences to exploding the bomb. You must be careful!

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.

每次你的炸弹爆炸，它都会通知bomblab服务器，你在实验室的最终分数中会损失1/2分（最多20分），所以爆炸炸弹会有后果。你一定要小心！

前四个阶段各得10分。第五阶段和第六阶段比较困难，所以他们各得15分。所以你能得到的最高分数是70分。

虽然阶段逐渐变得更难化解，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好的学生，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,

炸弹忽略空白输入行。例如，如果你用命令行参数运行炸弹，

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.

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.

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.  
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.

然后它将从psol.txt直到到达EOF（文件结束），然后切换到stdin。在一个脆弱的时刻，邪恶博士添加了这个功能，这样你就不必再为你已经消除的阶段重新输入解决方案了。

为了避免意外引爆炸弹，您需要学习如何单步执行汇编代码以及如何设置断点。您还需要学习如何检查寄存器和内存状态。做这个实验的一个好的副作用是你会很好地使用调试器。这是一项至关重要的技能，它将给你的职业生涯带来丰厚的回报。

## Logistics

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。所有的手柄都是电子的。澄清和更正将张贴在课程留言板上。

## Handin

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:

没有明确的移交。炸弹会自动通知你的教练你的进度，当你在它的工作。你可以通过查看班级记分板来了解自己的表现：

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，以显示每枚炸弹的进度。

**Hints *(Please read this!)***

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 technique, 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种。您可以在不运行程序的情况下对其进行详细的检查，并确切地了解它的作用。这是一种有用的技巧，但并不总是容易做到的。您也可以在调试器下运行它，一步一步地观察它做什么，并使用这些信息来消除它。这可能是最快的化解方法。

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 several reasons:

我们只提出一个要求，请不要使用暴力！你可以编写一个程序，尝试所有可能的键来找到正确的键。但这并不好，原因如下：

* You lose 1/2 point (up to a max of 20 points) every time you guess incorrectly and the bomb explodes.
* 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.
* 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 letters, then you will have 2680 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.
* •每次你猜错了，炸弹爆炸，你就会损失1/2分（最多20分）。
* •每次你猜错了，就会向bomblab服务器发送一条消息。您可能很快就会让这些消息充斥网络，并导致系统管理员撤销您的计算机访问权限。
* •我们没有告诉您字符串的长度，也没有告诉您字符串中有哪些字符。即使你做了（不正确的）假设，它们的长度都小于80个字符，并且只包含字母，那么每个阶段都会有2680个猜测。这将需要很长时间才能运行，而且在作业到期之前，您将无法得到答案。

There are many tools which are designed to help you figure out both 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.

有很多工具都是用来帮助你弄清楚程序是如何工作的，以及当程序不工作时有什么问题。以下是一些工具的清单，你可能会发现有助于分析你的炸弹，并提示如何使用它们。

* 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.

•gdb公司

GNU调试器，这是几乎所有平台上都可用的命令行调试器工具。您可以逐行跟踪程序，检查内存和寄存器，同时查看源代码和汇编代码（我们不会为您提供大多数炸弹的源代码）、设置断点、设置内存监视点以及编写脚本。

The CS:APP web site

<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，可以打印出来用作参考。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.
* 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.

–为了防止每次输入错误的输入时炸弹爆炸，您需要学习如何设置断点。

–对于联机文档，请在gdb命令提示符处键入“help”，或在Unix提示符下键入“man-gdb”或“info-gdb”。有些人还喜欢在emacs中以gdb模式运行gdb。

* 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及其地址。通过查看函数名，您可能会学到一些东西！

* 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.

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:

用这个来分解炸弹里的所有代码。你也可以只看单个函数。阅读汇编代码可以告诉你炸弹是如何工作的。

虽然objdump-d提供了很多信息，但它并不能告诉您整个过程。对系统级函数的调用以一种神秘的形式显示。例如，对sscanf的调用可能显示为：

8048c36: e8 99 fc ff ff call 80488d4 <\_init+0x1a0>

To determine that the call was to sscanf, you would need to disassemble within gdb.

要确定调用的是sscanf，您需要在gdb中进行反汇编。

* strings

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.

找一个特别的工具？文件如何？别忘了，命令、命令、人和信息是你的朋友。尤其是man-ascii可能会有用。信息气体将给你比你想知道的更多关于GNU汇编程序。此外，网络也可能是信息的宝库。如果你被难住了，可以向你的导师寻求帮助。