Bomb-lab 实验介绍

点击下载 cmu 官网实验说明详情

这个实验帮助学生练习和掌握机器级程序的基本原理，以及一些通用的调试和反向工程技巧。

二进制炸弹

实验中，一个“二进制炸弹”是指一个 Linux 下的可执行程序，包含六个关口。每个关口需要学生从控制台（stdin）输入一个猜测的特定的字符串（可以理解为拆弹密钥）。如果猜对了，则拆弹成功；如果猜错了，则炸弹爆炸并输出“BOOM!!!”。学生的目标是尽可能多次拆弹成功。

每个炸弹关口从不同角度考察机器语言程序：

- 第 1 关：字符串比较
- 第 2 关：循环
- 第 3 关：条件/开关
- 第 4 关：递归调用和栈约定
- 第 5 关：指针
- 第 6 关：链接表/指针/结构类型
- 越往后的关口，炸弹越难拆除。并且，如果学生在第 4 关的密码后面加上一个特定字符串的话，还可以开启一个隐藏的副本关口(提示：查看 phase_defused 和 secrete_phase 的代码)。

每个关口有三个不同的副本：‘a’，‘b’， 和‘c’。每个学生在每个关口会获得一个随机副本。并且，绝大部分关口副本都会在创建时引入一个随机常量，保证每个学生获得一个唯一炸弹，必须独立完成拆弹。

拆弹

在拆弹过程中，学生需要使用调试器，比如 gdb 或者 ddd，来反汇编二进制并且单步执行二进制码，通过理解每条汇编指令的用意来推测拆弹密码。拆弹成功会加分，失败会扣分。

自动评分系统

学生从服务器下载自己唯一的炸弹程序，在拆弹过程中，每次拆弹成功或者失败都会实时传回给服务器，并在服务器上的计分板页面实时反映当前炸弹的结果。

实验环境说明

平台与语言

- Linux
- C 语言、汇编语言

可能有用的工具

- Gdb GNU 发布的调试工具。为了从炸弹程序中找出触发炸弹爆炸的条件，可以借助 gdb 来对二进制程序进行分析。
  - 加载、启动待调试的二进制文件。
  - 设置断点。
  - 查看反汇编代码、程序变量、寄存器、栈内容等。
  - 动态改变程序的执行环境，如修改变量的值。
- objdump 查看二进制文件的符号表（包括函数、全局变量的名称和地址） 查看二进制文件的反汇编。
- strings 显示二进制文件中的可打印字符串。

实验数据如何获取？

从 指定网站 获取实验数据压缩包，需要输入学号和邮箱地址。提交后，会自动下载该学号对应的数据包。

说明：

每个请求的实验数据包绑定到一个特定的学号，实验进行过程中会自动计算该学号的得分，所以请务必使用自己的数据包。

如果实验数据包丢失，可以重新输入学号和邮箱地址获取。

实验数据

实验数据包为一个压缩文件 bomb1.tar。解压缩后，里面有三个文件：

- bomb：二进制可执行炸弹程序。
- bomb.c：二进制炸弹对应的 main 函数的内容，可以看成一个辅助拆弹过程的提示。
- README：说明该实验数据包绑定到哪位学生。

实验内容

实验步骤提示

```
objdump -d bomb > disassemble.txt
```

上述命令可以获得二进制炸弹 bomb 的反汇编代码。通过查看反汇编文件中的 main 函数，可以找到对应第一个关卡的函数名。接下来在反汇编文件中找到该函数对应的代码。在其中可以找到对应的指令。对于第一关（字符串比较）来说，是一个字符串相关的函数调用指令（call strings_not_equal）。利用第三章的知识，确定参与比较的两个参数的存储位置。猜测是一个存在代码中，一个通过从标准输入读取（即学生输入的密码字符串）。通过找到代码中存储的字符串，就可以推测出第一关的密码。

实验结果的提交

如果学生在拆弹过程中保持联网，则不需要提交实验结果。学生在拆弹过程中，bomb 程序每次发生拆弹成功或失败都会将信息反馈到服务器记录，并实时更新当前得分状况。查看实时得分状况，点击这里。

郑重说明

凡是用学号领取的实验，服务器上都有记录，每个同学唯一。不要使用别人的炸弹，否则服务器进行答案比较时，会算成 0 分。因为这种行为系作弊行为，拒绝申诉和二次考核。