

综合实验 二进制炸弹实验

一、实验目的

本实验通过要求你使用课程所学知识拆除一个“binary bombs”来增强对程序的机器级表示、汇编语言、调试器和逆向工程等方面原理与技能的掌握。一个“binary bombs”（二进制炸弹，下文将简称为炸弹）是一个 Linux 可执行程序，包含了 6 个阶段（或层次、关卡）。炸弹运行的每个阶段要求你输入一个特定字符串，你的输入符合程序预期的输入，该阶段的炸弹就被拆除引信即解除了，否则炸弹“爆炸”打印输出 “BOOM!!!”。

二、实验内容

拆除尽可能多的炸弹层次，每个炸弹阶段考察了机器级程序语言的一个不同方面，难度逐级递增：

阶段 1：字符串比较

阶段 2：循环

阶段 3：条件/分支

阶段 4：递归调用和栈

阶段 5：指针

阶段 6：链表/指针/结构

另外还有一个隐藏阶段，只有当你在第 4 阶段的解后附加一特定字符串后才会出现。为完成二进制炸弹拆除任务，你可以使用 gdb 调试器、objdump 等工具来反汇编炸弹的可执行文件并单步跟踪调试每一阶段的机器代码，从中理解每一汇编语言代码的行为或作用，进而设法推断拆除炸弹所需的目标字符串。比如在每一阶段的开始代码前和引爆炸弹的函数前设置断点。

三、实验程序

你将在下面网站获得你的 bomb (<http://cslabcms.nju.edu.cn/bomb/>)

在这里你将看到一个二进制炸弹请求框，你可以填入你的学号然后按下下载按钮。这个服务器将返回给你的浏览器一个*.tar.gz文件的bomb，*代表你的学号。

Tar文件包含2个文件：

Bomb:可执行的64位二进制bomb

Bomb.c:写有bomb的主程序的源文件

你的任务是去拆除炸弹，你可以用许多工具去帮助你拆除你的炸弹。最好的方法是去用你最喜欢的调试器去调试你的二进制程序，如 gdb。（感兴趣的同学可以学习使用 cgdb 最新版本来进行调试分析 <https://cgdb.github.io/>）

Bomb 程序可以从 stdin 或者文件进行读入。建议通过文件进行读入，文件中的每一行包含一个输入字符串。

四、实验样例

实验成功时的输出：

```
harry@harry-VirtualBox:~$ ./bomb solution
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Phase 1 defused. How about the next one?
That's number 2. Keep going!
Halfway there!
So you got that one. Try this one.
Good work! On to the next...
Curses, you've found the secret phase!
But finding it and solving it are quite different...
Wow! You've defused the secret stage!
Congratulations! You've defused the bomb!
```

其中 solution 文件可供参考的样例为：.

```
star
1 2 3 1 2 3
0 61
6 xyz(secret phrase)
1<1000
2 5 4 6 3 1
7
```

五、提交要求

1、需要提交实验报告。实验报告内注明姓名学号，报告的文件名为自己的学号，在报告内要写明自己的分析过程，每个阶段都需要有一个详细的分析过程。

2、需要提交如第四节中所示的 solution 文件。为了避免出现不必要的麻烦，在提交 solution 文件前建议执行 ./bomb solution 对答案进行测试。