

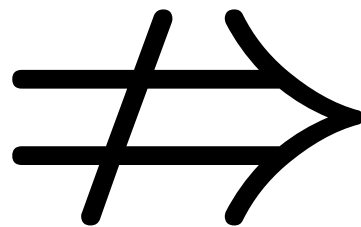
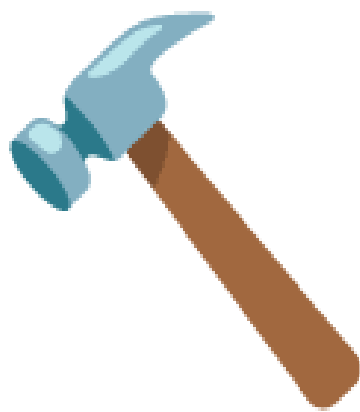
实验导引 1 / bomblab

大家都能连到服务器上了嘛

连不上的同学们尽快找 TA 真人快打

背景

bomblab = bomb lab = /bɒm læb/, 来自 CSAPP



实验简述

你将得到一个可执行文件（炸弹），共有多个阶段，每个阶段可以通过输入密码拆除。你的任务是找到这样的一个输入，使得炸弹通过所有阶段的检查，不被引爆。

登录服务器，访问 `~/bomblab` 目录，将会有三个文件：可执行文件 `bomb`，**部分**源代码 `bomb.c`，以及一个空的输入文件 `input.txt`

实验简述

你将得到一个可执行文件（炸弹），共有多个阶段，每个阶段可以通过输入密码拆除。你的任务是找到这样的一个输入，使得炸弹通过所有阶段的检查，不被引爆。

登录服务器，访问 `~/bomblab` 目录，将会有三个文件：可执行文件 `bomb`，**部分**源代码 `bomb.c`，以及一个空的输入文件 `input.txt`

```
./bomb          # 使用标准输入  
./bomb input.txt # 使用 input.txt 的内容作为输入，之后使用标准输入
```

实验工具

- GDB 将是本实验主要的工具。VSCode 内的 GDB 已经被配置完成（使用 `input.txt` 作为输入）¹
- `objdump` 可以作为静态分析的工具
- `xxd` 可以显示任意文件的 Hex dump，并且将可以解读为可见 ASCII 字符的内容显示出来，可以配合 `objdump` 进行静态分析。
- `strings` 可以将二进制文件中所有可见字符组成的 C 字符串显示出来。

¹我们注意到 VSCode GDB 调试器在缺乏函数源代码时，Disassembly view 中有时光标位置显示错误。步进等控制操作是正确的。

实验要求

- 你需要提交输入和一个实验报告，评分占比及提交方式：
 - 60%: 请将你的输入填入服务器上的 `~/bomblab/input.txt`。共六个非隐藏阶段，每个 10%，隐藏阶段不做要求
 - 40%: 实验报告请通过网络学堂提交，简述你破解炸弹每个阶段的方式
- 每个同学得到的炸弹都是独一无二的
- 实验截止日期：11 月 25 日（星期一） 23:59
 - 迟交 Multiplier: $\max(60\%, \min(80\%, 80\% - \text{迟交天数} * 1\%))$
- 预估时间：15-30h。Start early!

Help, I'm stuck!

- GDB quick reference by UT Austin
<https://users.ece.utexas.edu/~adnan/gdb-refcard.pdf>
- System V ABI
https://wiki.osdev.org/System_V_ABI
- 看函数名猜功能
- 询问同学/助教

Q&A

~~Happy blowing up!~~ Good luck with the bomb!