**《软件安全》实验报告**

姓名：胡博浩 学号：2212998 班级：信息安全

**一、实验名称：**

Angr应用示例

**二、实验要求：**

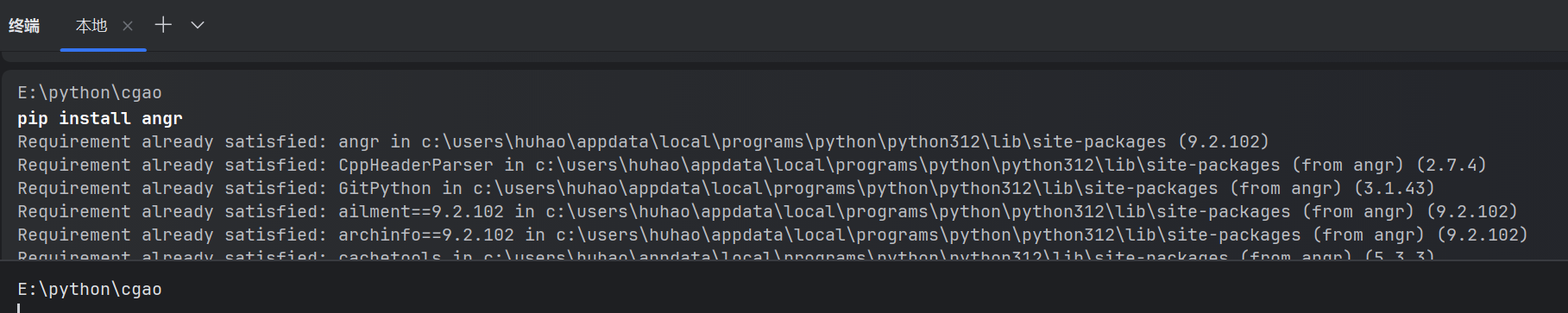
根据课本8.4.3章节，复现sym-write示例的两种angr求解方法，并就如何使用angr以及怎么解决一些实际问题做一些探讨。

**三、实验过程：**

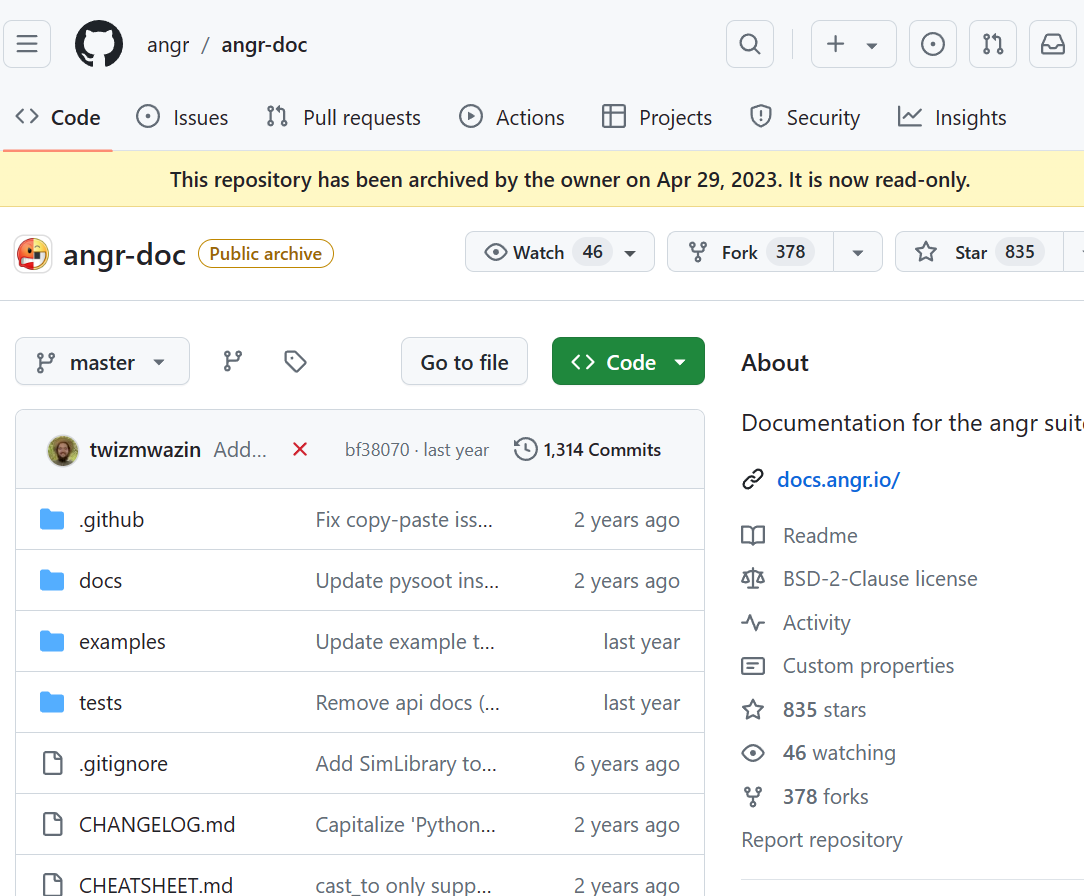
（一）复现实验

1.安装Angr：

在pycharm的终端中使用命令：pip install angr（图中我已安装好Angr）



在github上下载Angr的官方文档

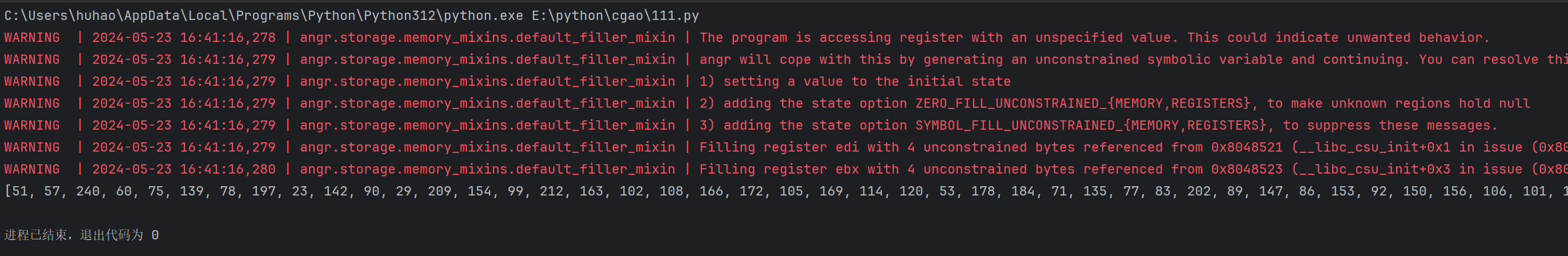


2.使用example中的样例源代码求解问题：

在angr-doc里有各类Example，展示了Angr的用法，我们以sym-write为例子，来说明angr的用法。代码如下：

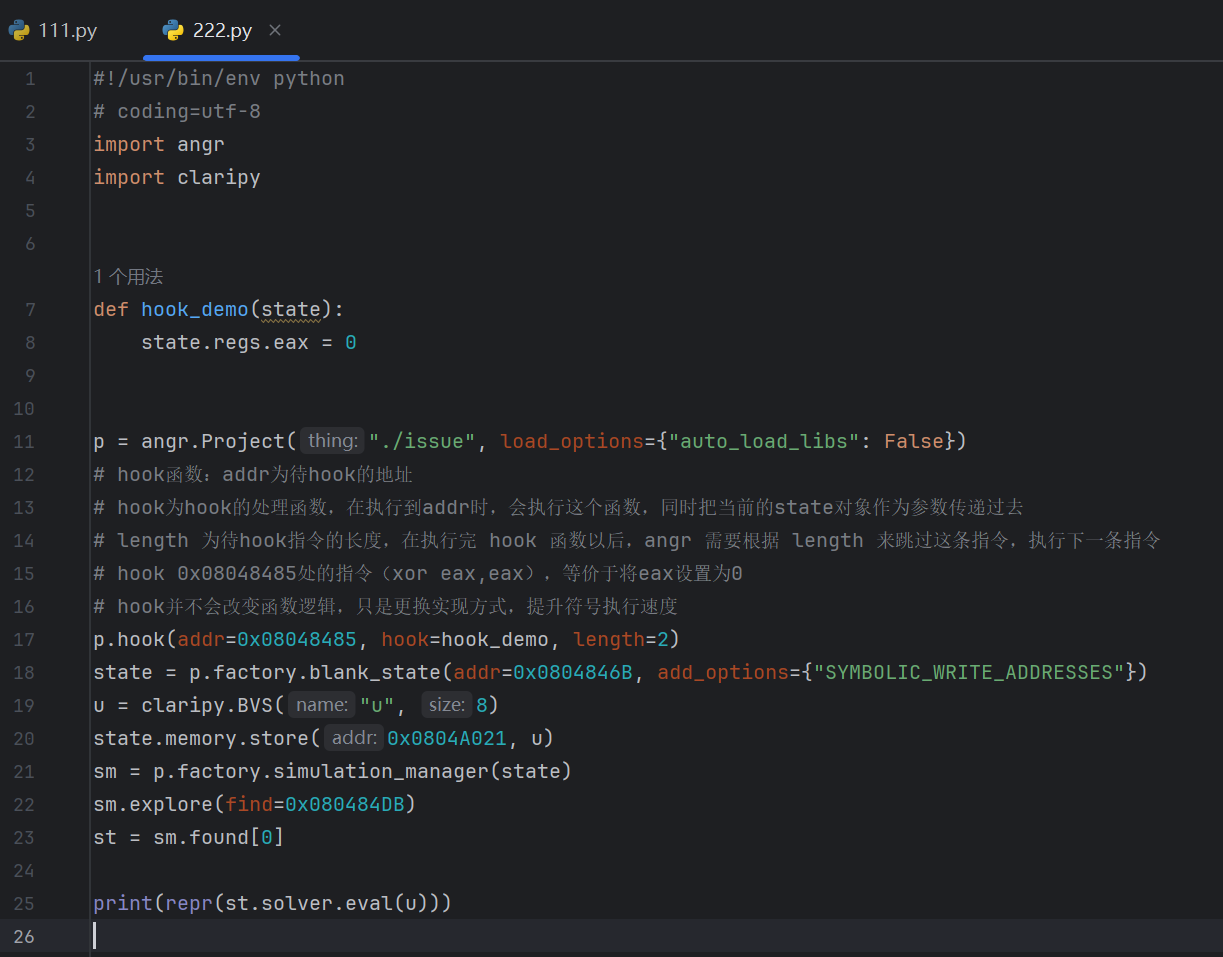


运行代码，得到如下结果：



3.使用实验指导书中的给出的第二种方法，借助hook求解问题。

代码如下：



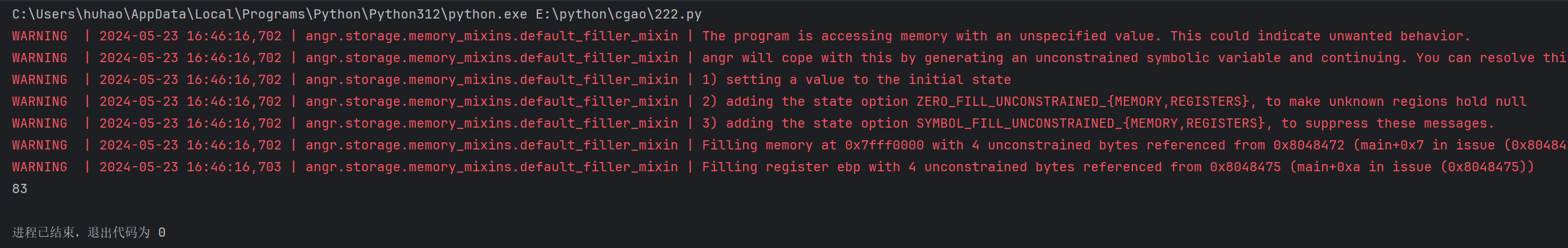
上述代码与前面的解法有三处区别：

（1）采用了hook函数，将0x08048485处的长度为2的指令通过自定义的hook\_demo进行替代，功能是一致的，原始xor eax, eax和state.regs.eax = 0是相同的作用，这里只是演示，可以将一些复杂的系统函数调用，比如printf等，可以进行hook，提升符号执行的性能。

（2）进行符号执行得到想要的状态，有变化，变更为find=0x080484DB。因为源程序win和lose是互斥的，所以，只需要给定一个find条件即可。

（3）最后，eval(u)替代了原来的eval\_upto，将打印一个结果出来。

运行代码，结果如下：



（二）如何使用angr以及怎么解决一些实际问题

angr是一个功能强大的二进制代码分析工具，能够自动化分析二进制文件并找出潜在漏洞。它基于Python，集成了多种分析技术，能够执行动态符号执行和多种静态分析。以下是使用angr的主要步骤和在实际应用中的一些探讨。

* 如何使用

1.新建一个工程并导入二进制文件：

创建一个angr项目并导入目标二进制文件。在这个过程中，可以通过设置参数选项来决定是否自动加载依赖项。通常，为了加快分析速度和减少不必要的复杂性，可以选择禁用自动加载库文件。

2.初始化SimState对象：

SimState对象用于模拟程序的运行状态，包括程序内容、寄存器、文件系统数据和符号信息等。通过以下两种方式初始化SimState对象：

（1）blank\_state()：通过指定起始地址来创建状态。

（2）entry\_state()：创建程序在初始运行时的状态。

可以通过add\_option()方法为状态添加独立选项，以定制分析行为。

3.创建符号变量：

符号变量用于替代具体输入值，使angr能够探索不同输入条件下的程序行为。创建符号变量后，将其保存到指定内存地址（例如，二进制文件的.bss段）。

4.创建Simulation Manager对象：

Simulation Manager对象与初始化的状态相关，管理和控制符号执行过程。它可以跟踪程序执行路径，并在遇到条件分支时分裂状态进行探索。

5.使用explore函数对状态进行搜索：

使用Simulation Manager的`explore`函数执行符号执行搜索。定义成功状态和避免状态的条件。angr会根据这些条件进行路径探索，并试图找到满足条件的执行路径。

6.获取结果：

找到满足条件的状态后，通过solver求解器获取符号变量的具体值。solver求解器会解析符号变量的约束，得出满足条件的实际输入。

* 实际应用

angr可以用于多种实际应用场景，包括破解程序输入校验、漏洞挖掘和复杂逻辑分析。

1.破解程序输入校验：

对于需要特定输入才能通过校验的程序，通过angr可以自动化地寻找能够通过校验的输入。angr会探索所有可能的输入组合，并找到能够使程序输出预期结果的输入。

2.漏洞挖掘：

angr可以自动检测常见的漏洞类型，例如缓冲区溢出、格式化字符串漏洞等。通过符号执行和状态监控，angr能够发现程序中可能导致这些漏洞的输入条件。

3.结合静态分析和符号执行：

可以先使用静态分析工具（如IDA Pro）确定程序中的关键函数或代码段，然后使用angr进行符号执行以深入分析这些部分。生成程序的控制流图（CFG）后，可以识别感兴趣的函数，并从这些函数开始进行符号执行，精确地分析程序行为。

**四、心得体会：**

通过这次实验，我对使用Angr工具进行符号执行和二进制分析有了深入的理解和实践体验。这次实验不仅增强了我的技术能力，还让我在实际应用中感受到了符号执行的强大之处和二进制分析的复杂性。

在实验中，通过使用Angr进行符号执行，我学会了如何将程序的输入表示为符号变量，并跟踪这些变量在程序执行过程中的变化。通过查阅其文档，也学习了一些常用方法和基本函数的API。通过阅读其求解代码，对于Angr的基本逻辑也有了比较明确的认识。符号执行的魅力在于，它可以探索程序所有可能的执行路径，而不仅仅是依赖于特定的输入。这种全面的分析方法让我对程序的行为有了更全面的认识。