《漏洞利用及渗透测试基础》实验报告

姓名：姜志凯 学号：2011937 班级：信安班

**实验名称：**

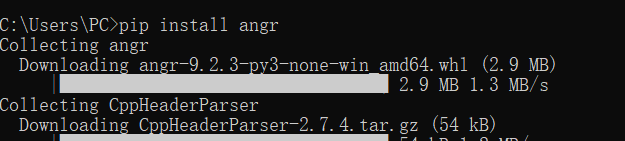
Angr应用示例

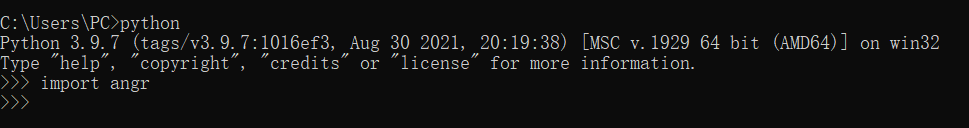
**实验要求：**

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。

**实验过程：**

1. Windows下安装Angr，首先安装python3，然后再命令行输入pip install angr命令安装Angr，并对安装结果进行测试，进入python界面，输入import angr，不报错则安装成功。





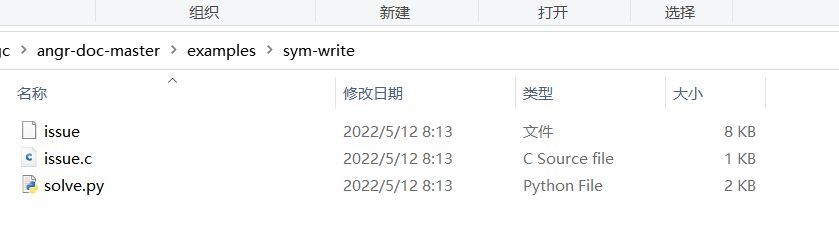
现在就可以在python的可视化界面中直接import angr了。

1. Angr示例：sym-write

issue.c程序通过输入u的值，根据u的值不同，程序输出“you win！”or“you lose！”。

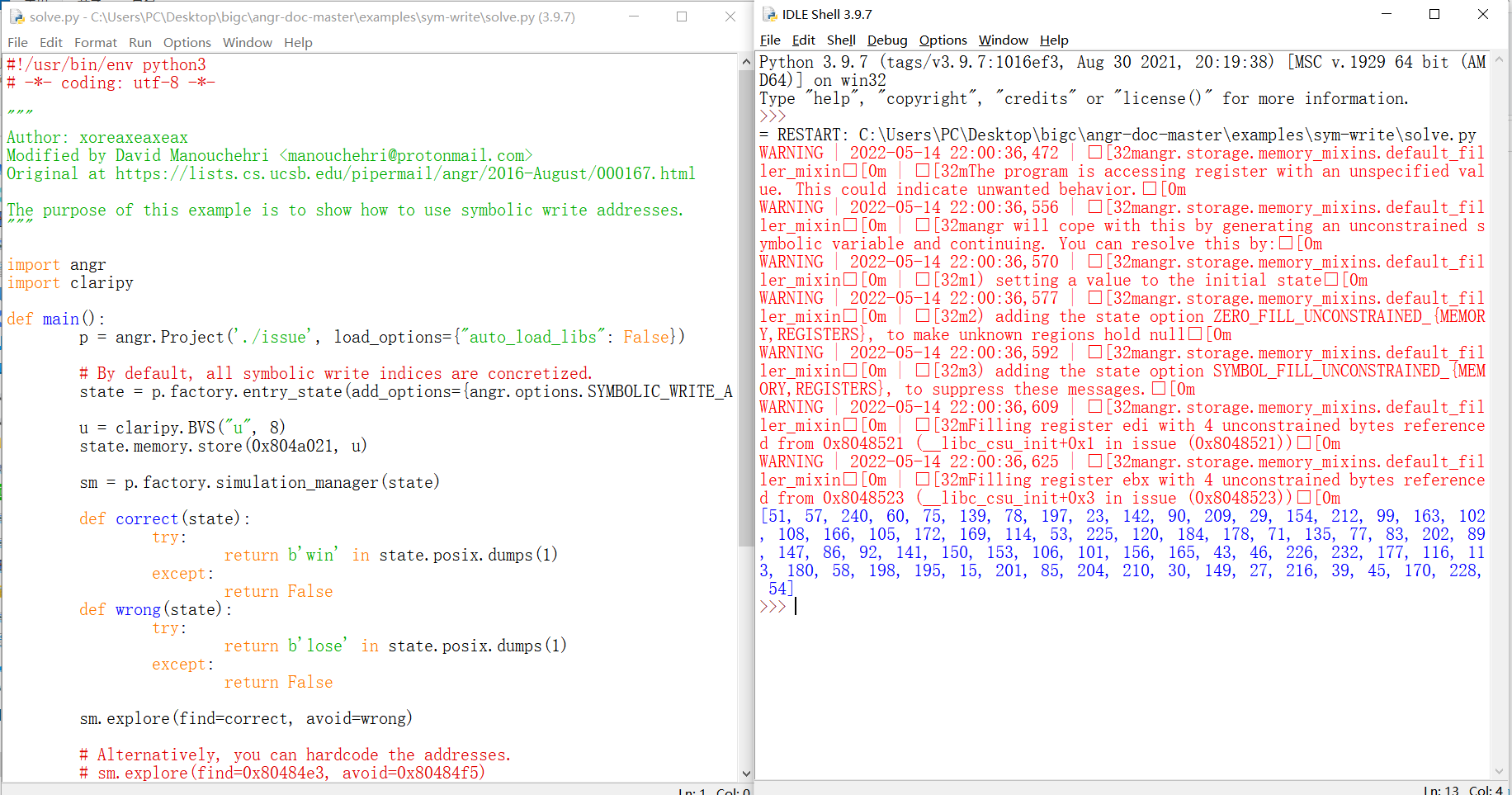
利用Angr，经过变量符号化->动态符号执行->获取路径约束条件->约束求解过程，得到所有符合条件的u值。

下载的zip中的example中有该示例



其中issue.c为测试程序源码，issue为其二进制文件，solve.py为利用Angr求解u值的程序，Angr用的是源程序的二进制文件，所以程序里要获取issue文件

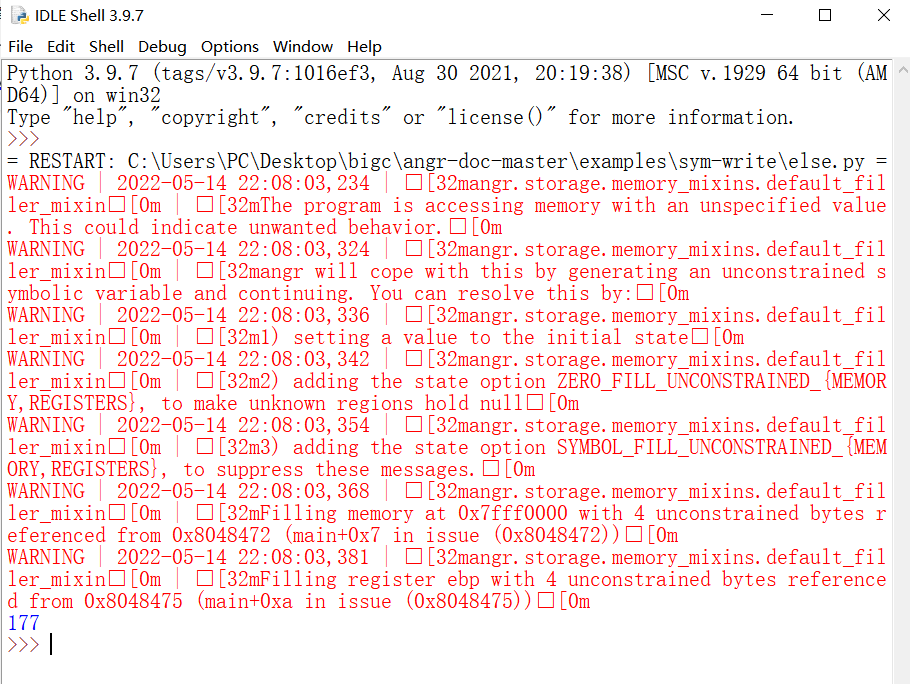
用python3.9.7打开solve.py，然后运行



这里的每一个u值输入到源程序，都能得到“you win！”，成功！

1. 另一种解法

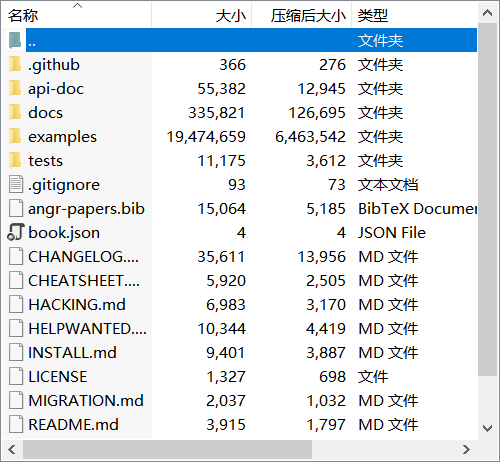




**心得体会：**

1. GitHub上有angr的开源项目https://github.com/angr以及相关的文档信息，https://github.com/angr/angr-doc里有各类Example，展示了Angr的用法，比如cmu\_binary\_bomb、simple\_heap\_overflow等二进制爆破、堆溢出等漏洞挖掘、软件分析的典型案例，可以进行拓展。

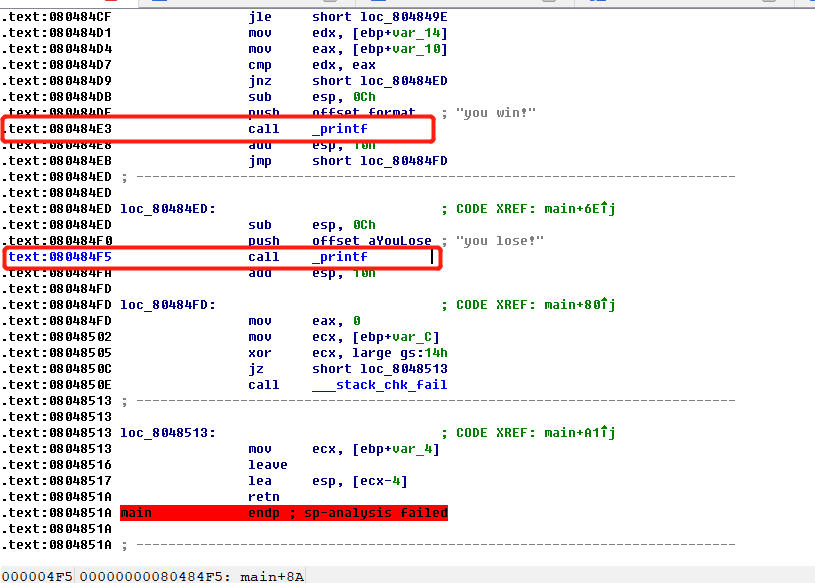




1. Angr是一个二进制代码分析工具，能够自动化完成二进制文件的分析，并找出漏洞。

Angr是一个基于python的二进制漏洞分析框架，它将以前多种分析技术集成进来，它能够进行动态的符号执行分析（如KLEE和Mayhem），也能够进行多种静态分析。

1. 上述求解过程的关键步骤：（1）创建一个Simulation Manager对象，管理运行得到的状态对象；（2）定义函数state.posix.dumps获得所有标准输出；（3）动态符号执行，得到想要的状态使用explore函数进行状态搜寻，也可以写成sm.explore(find=0x80484e3, avoid=0x80484f5)；约束求解，获得到state之后，通过solver求解器，求解u的值。



如图可见，80484e3和80484f5分别为两个输出的地址，用来限制求解条件，得到想要的结果。

1. 另一种方法区别之处：①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；②最后，eval(u)替代了原来的eval\_upto，将打印一个结果出来；③提供了一种思路，可以采用hook函数，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（这里只是演示一下这个功能，并没有起什么作用，因为hook替换函数与原来的功能是一致的）。

此外，通过本实验，掌握了多个汇编语言的用法