《漏洞利用及渗透测试基础》实验报告

姓名：赵悦蛟 学号：2313650 班级： 1071

**实验名称：**

Angr 应用示例

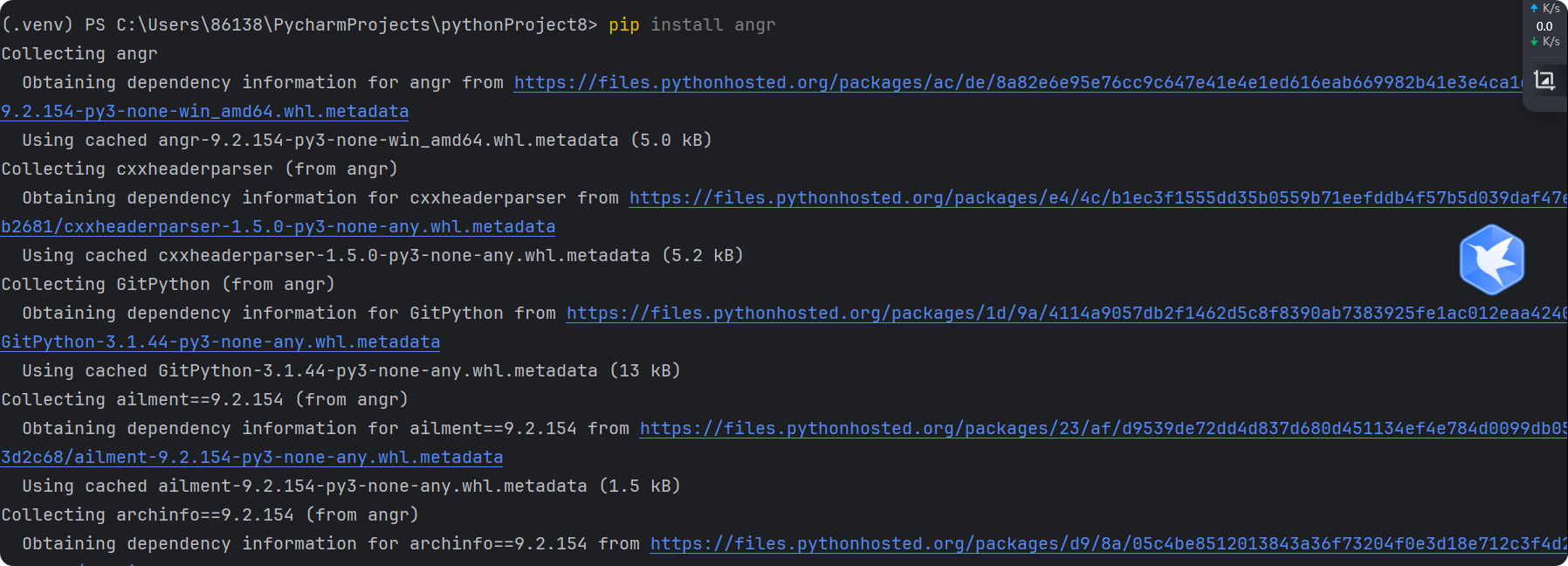
**实验要求：**

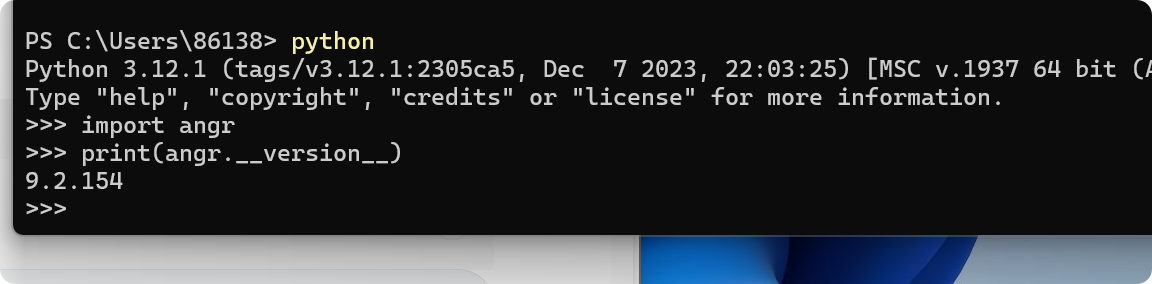
根据课本8.4.3章节，复现sym-write示例的两种angr求解方法，并就如何使用angr以及怎么解决一些实际问题做一些探讨。

**实验过程：**

1. 安装angr

在终端中通过pip install angr安装angr。





1. Angr的第一种解法

在github 上下载angr的官方文档。从 angr - doc 里获取示例代码，首先以 sym - write 为例，其包含源码issue.c和solve.py。

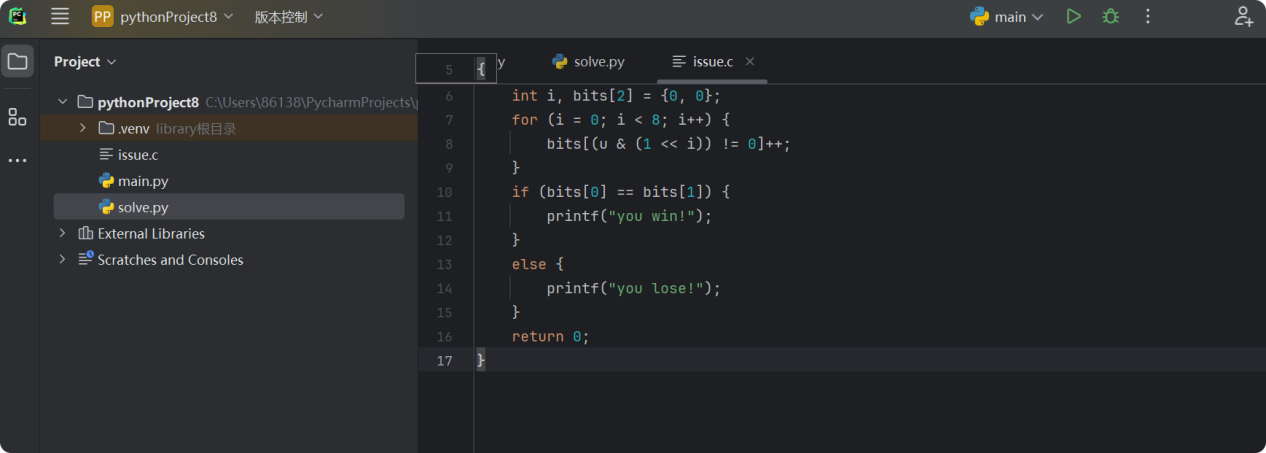
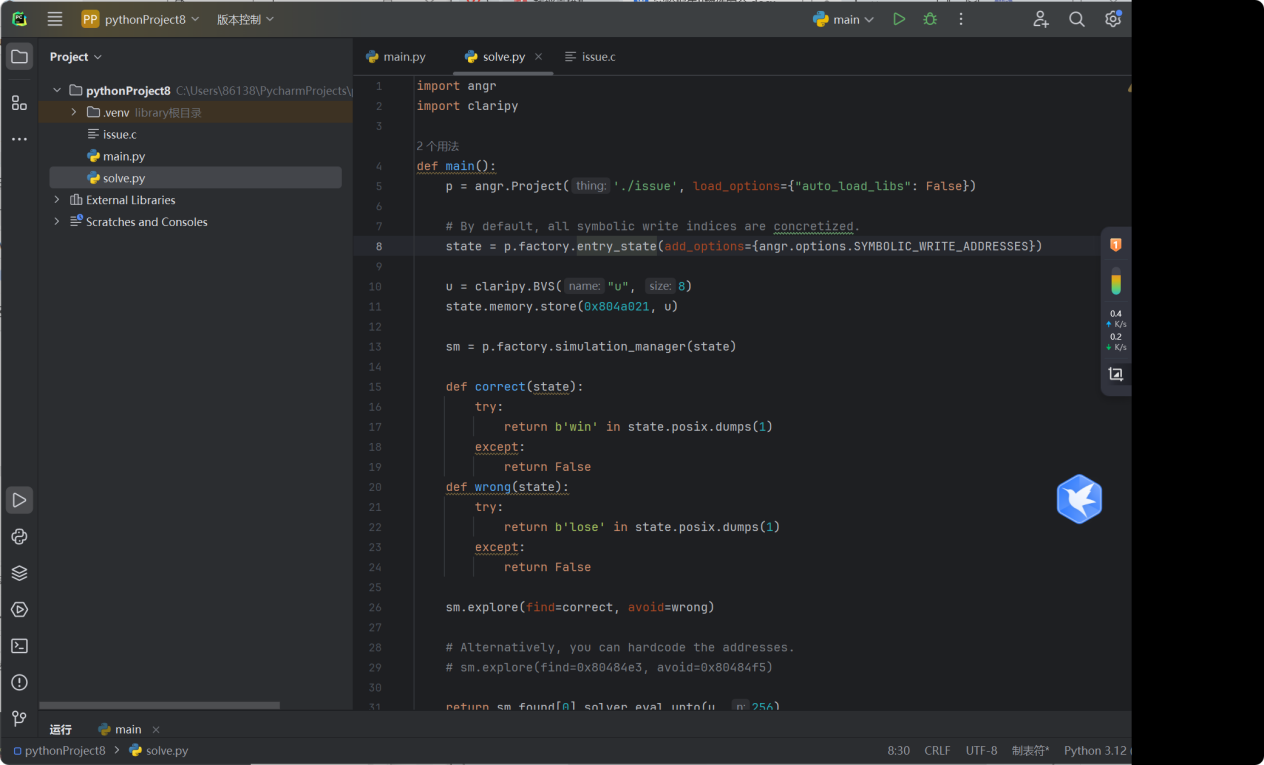


图 1isssue文件的代码

issue.c中定义了一个字符变量u并初始化为 0，在main函数里通过循环统计u二进制表示中 0 和 1 的个数，若两者相等则打印you win!，否则打印you lose!。



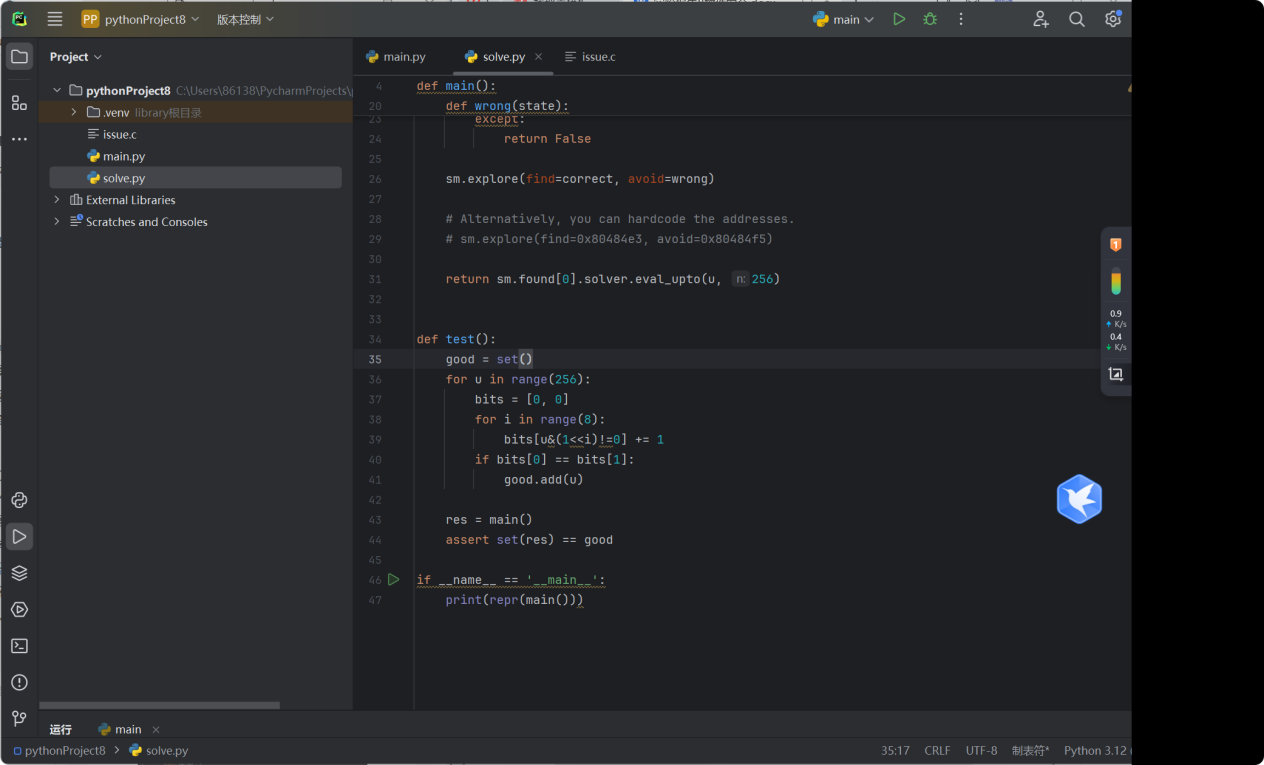


图 2solve.py

solve.py中，先导入angr和claripy库。在main函数里，首先创建angr.Project对象加载二进制文件issue，并设置auto\_load\_libs=False避免自动载入依赖库。

State=p.factory.entry\_state(add\_options={angr.options.SYMBOLIC\_WRITE\_ADDRESSES})使用p.factory.entry\_state初始化一个模拟程序状态的SimState对象state，并添加angr.options.SYMBOLIC\_WRITE\_ADDRESSES选项。接着将变量u符号化，使用state.memory.store将符号化变量u存储到指定内存地址。创建模拟管理器sm，定义correct和wrong函数来判断程序执行后的状态。通过sm.explore进行符号执行，查找满足correct函数且避免wrong函数的状态。也可以使用find=0x80484e3, avoid=0x80484f5替代函数判断，这两个地址分别对应printf("you win!")和printf("you lose!")的汇编语句。最后使用solver求解器的eval\_upto函数获取变量u的多个可能解

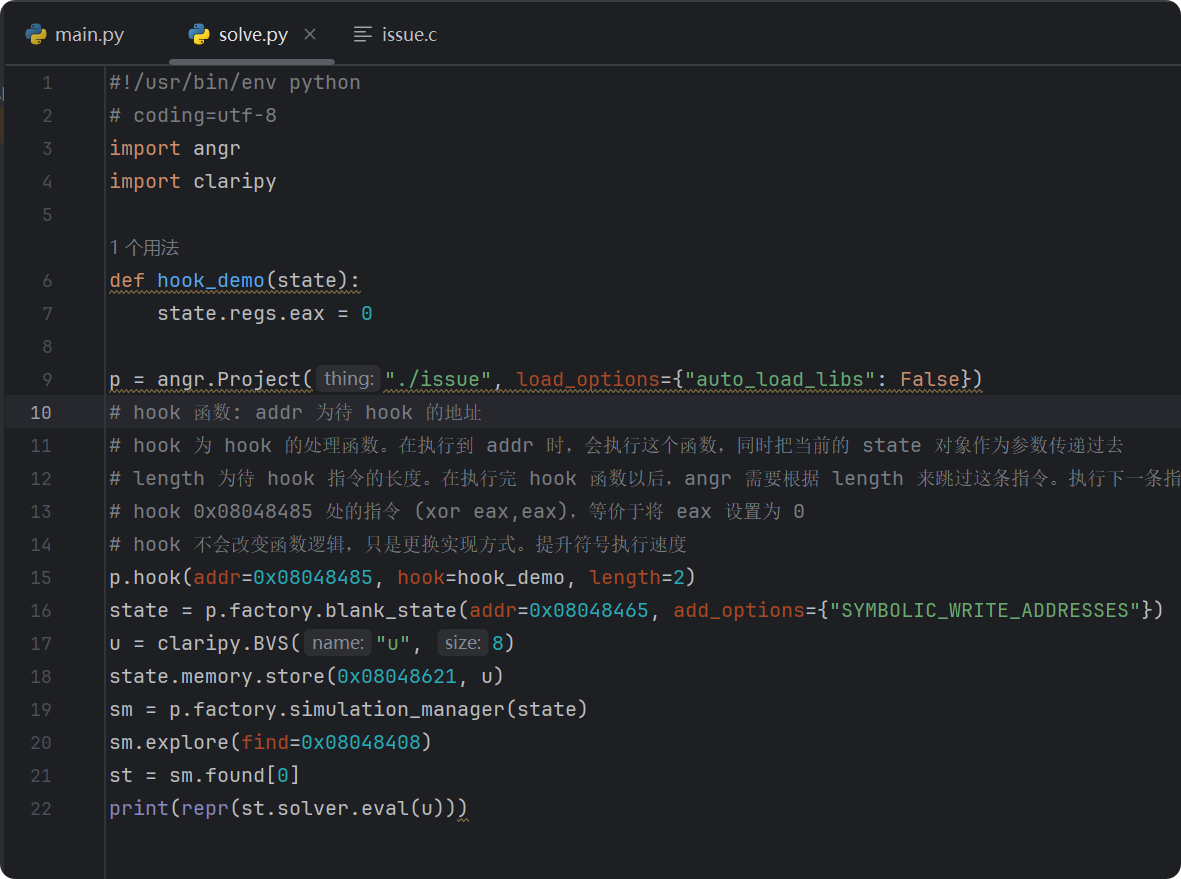
运行代码得到结果：



输出了u的结果。

1. Angr的第二种解法

使用第二种求解方式：



第二种方法定义hook\_demo函数，用于在特定地址进行 hook 操作，将eax寄存器设置为 0。创建angr.Project对象加载二进制文件，与第一种解法类似。使用p.hook对0x08048485处长度为 2 的指令进行 hook，执行hook\_demo函数。使用p.factory.blank\_state初始化模拟程序状态对象state，指定程序起始运行地址并添加SYMBOLIC\_WRITE\_ADDRESSES选项。将变量u符号化并存储到指定内存地址。通过sm.explore进行符号执行，设置find=0x080484DB查找满足条件的状态。最后使用solver求解器的eval函数获取变量u的一个解

运行结果如图：



**心得体会：**

在本次《漏洞利用及渗透测试基础》实验中，围绕 Angr 框架展开的 sym-write 示例复现，让我对二进制分析和漏洞利用有了更深入的认识。

在技术实践层面，我掌握了 Angr 框架的基本使用方法。

实验中的两种求解方法让我深入理解了符号执行和约束求解的原理及应用。第一种解法通过创建模拟程序状态、符号化变量、利用模拟管理器进行符号执行等操作，展示了如何在不实际运行程序的情况下，通过符号化手段探索程序路径并找到满足条件的输入值。这一过程让我理解了 Angr 框架的核心功能，也体会到符号执行在漏洞挖掘中的作用。例如，在判断程序输出结果时，我学会了如何通过操作模拟状态对象获取程序的输出信息，这对识别程序异常行为或潜在漏洞非常关键。

第二种解法引入的 hook 函数是一个重要概念。它展示了在符号执行过程中，可以通过自定义 hook 函数改变程序执行行为，优化执行效率。实验中通过对特定指令进行 hook 操作，简化了复杂操作的处理，让符号执行更高效地找到目标状态。这让我认识到在面对复杂二进制程序时，可以灵活运用 hook 技术对关键部分进行处理，提高分析效率。

这次实验也让我认识到二进制分析在网络安全领域的重要性。Angr 作为强大的二进制分析框架，为检测二进制文件中的漏洞提供了有效手段。通过学习使用 Angr 进行漏洞检测和利用，我对网络安全防护有了新的认识。