《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：**

AFL模糊测试实验

**实验要求：**

根据课本7.4.5章节，复现AFL在KALI下的安装、应用，查阅资料理解覆盖引导和文件变异的概念和含义。

**实验过程：**

1. 新建Linux虚拟机

下载Linux镜像，然后打开 VMware Workstation，创建一个新的虚拟机，安装 Ubuntu 系统



图 1安装Ubuntu系统

2.安装AFL

首先进行依赖安装，在终端中使用

sudo apt-get update

sudo apt-get install -y build-essential git clang llvm命令完成依赖安装

然后使用

git clone https://github.com/google/AFL.git

cd AFL

make

sudo make install

afl-fuzz --version命令下载AFL源码并进行编译和安装，输出版本信息，成功安装。

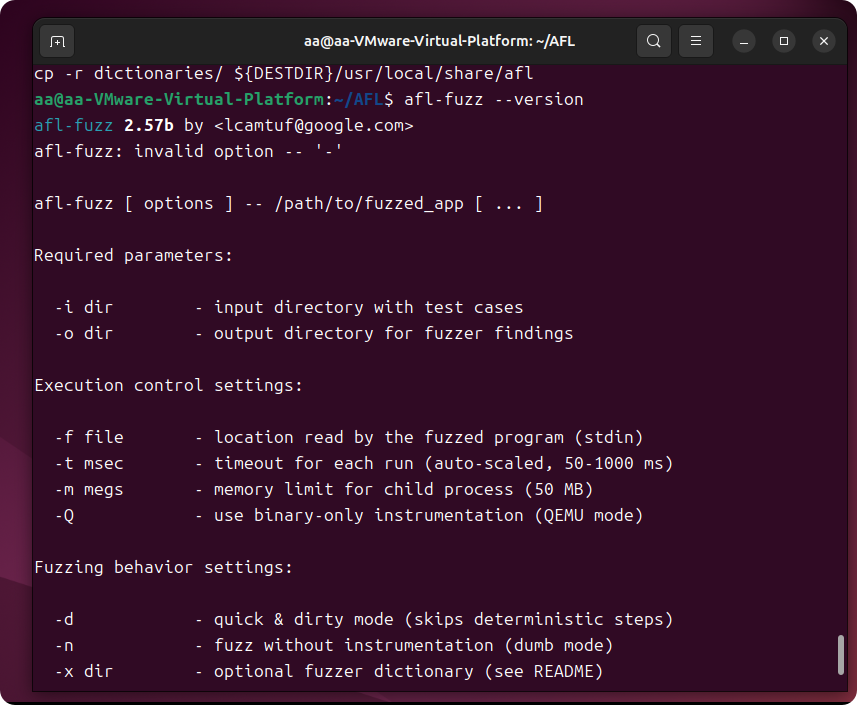


图 2安装AFL

使用echo core | sudo tee /proc/sys/kernel/core\_pattern

ulimit -c unlimited命令配置好系统环境。

3.AFL测试

AFL可以进行白盒测试也可以进行黑盒测试，本次实验中进行白盒模糊测试

1. 创建本次实验的程序

新建文件夹demo，并创建本次实验的程序test.c，该代码编译后得到的程序如果被传入deadbeef则会终止，如果传入其他字符会原样输出。

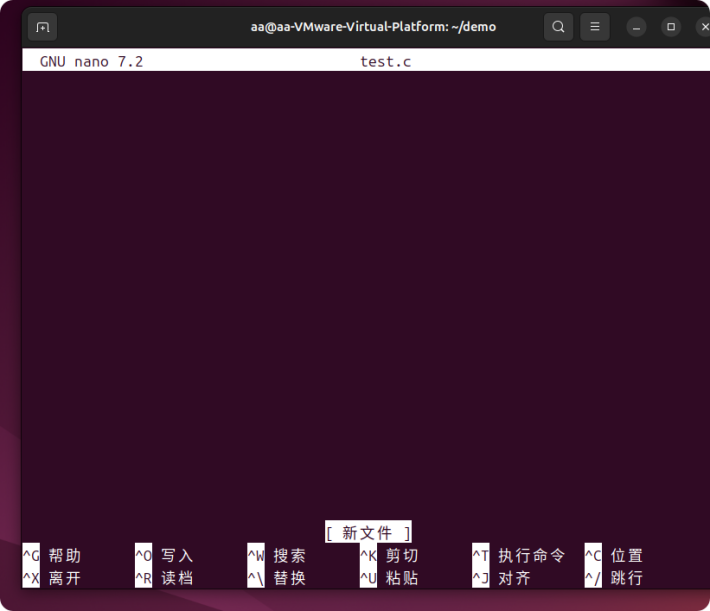


图 3创建test.c

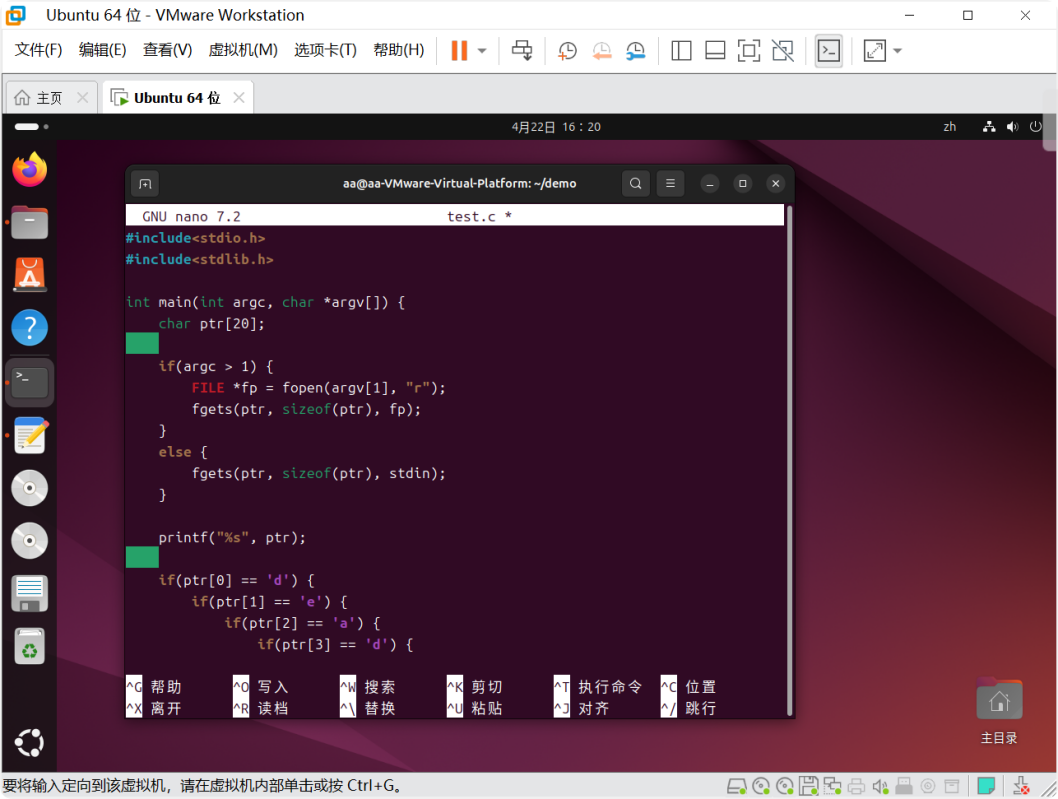


图 4编写实验中用的代码

使用AFL的编译器编译test文件，在终端中使用afl-gcc -o test test.c命令编译文件。

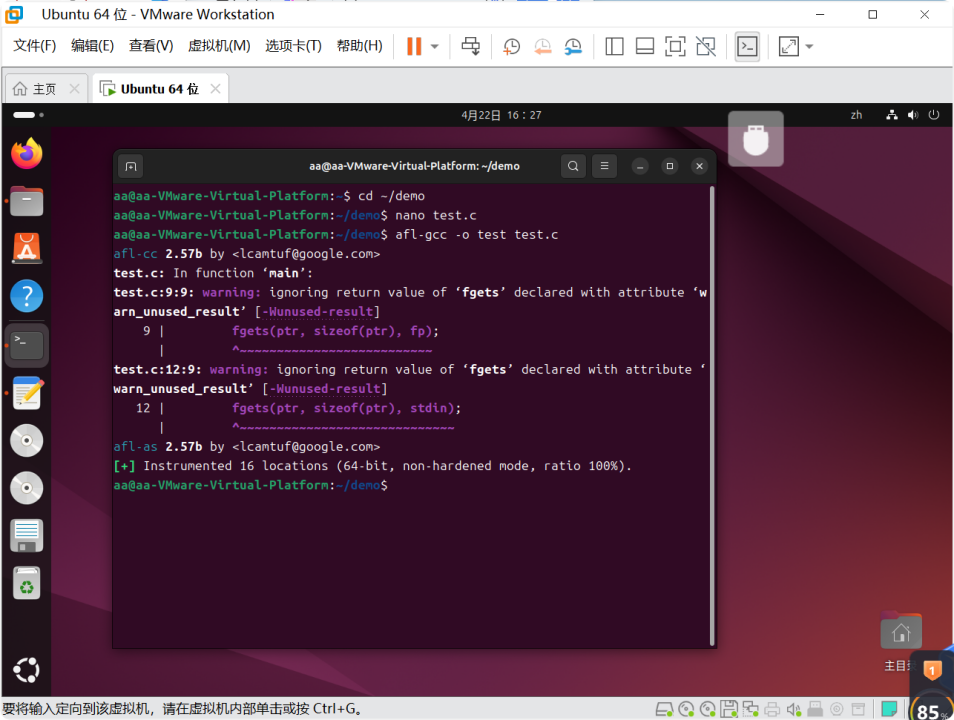
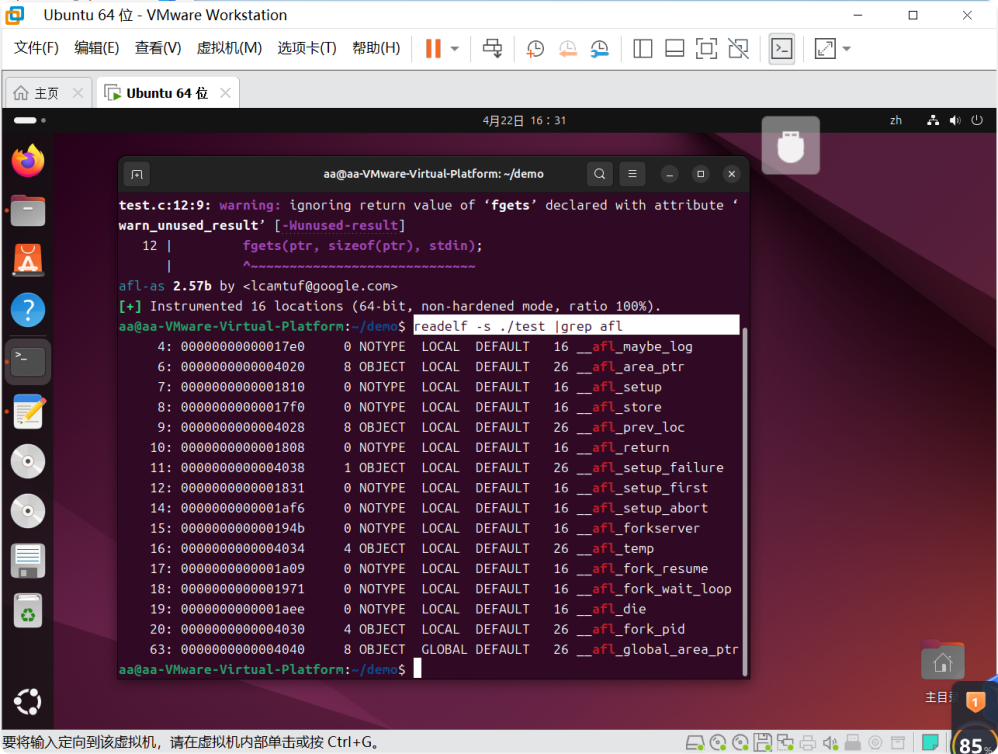
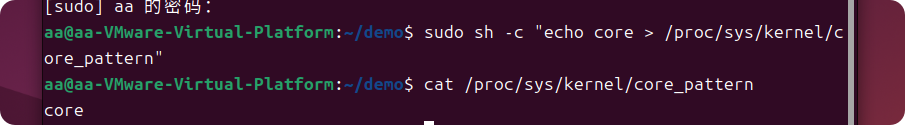


图 5编译程序后的结果

使用readelf -s ./test |grep afl命令查看编译后的插桩

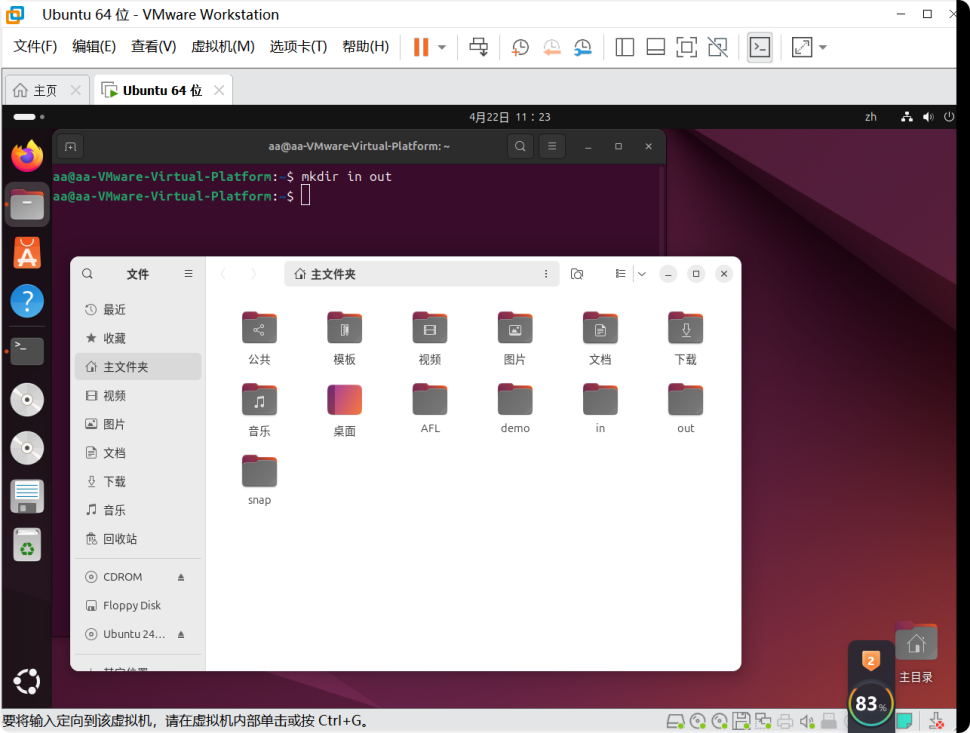


在进行下一步之前，使用echo core >/proc/sys/kernel/core\_pattern命令将coredumps输出为文件，而不是将他们发送到特定的崩溃处理程序。

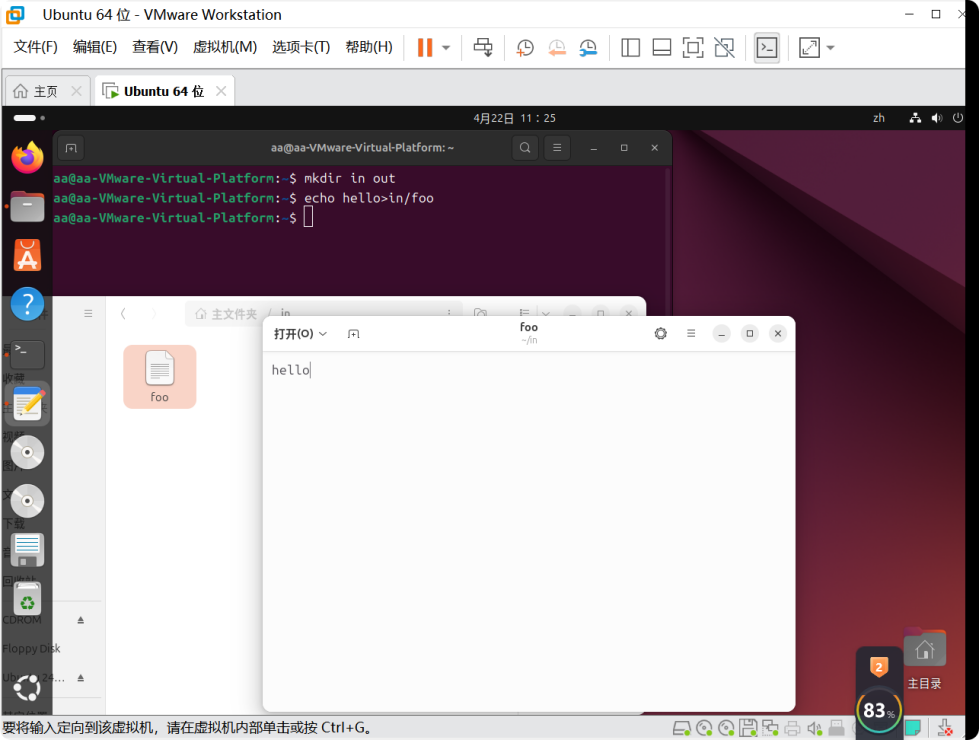


1. 创建测试用例

使用mkdir in out命令创建两个文件夹 in 和 out，分别存储模糊测试所需的输入和输出相关的内容。



使用：echo hello> in/foo命令在输入文件夹中创建一个包含字符串“hello”的文件。



1. 进行模糊测试

使用afl-fuzz -i in -o out -- ./test @@命令启动模糊测试

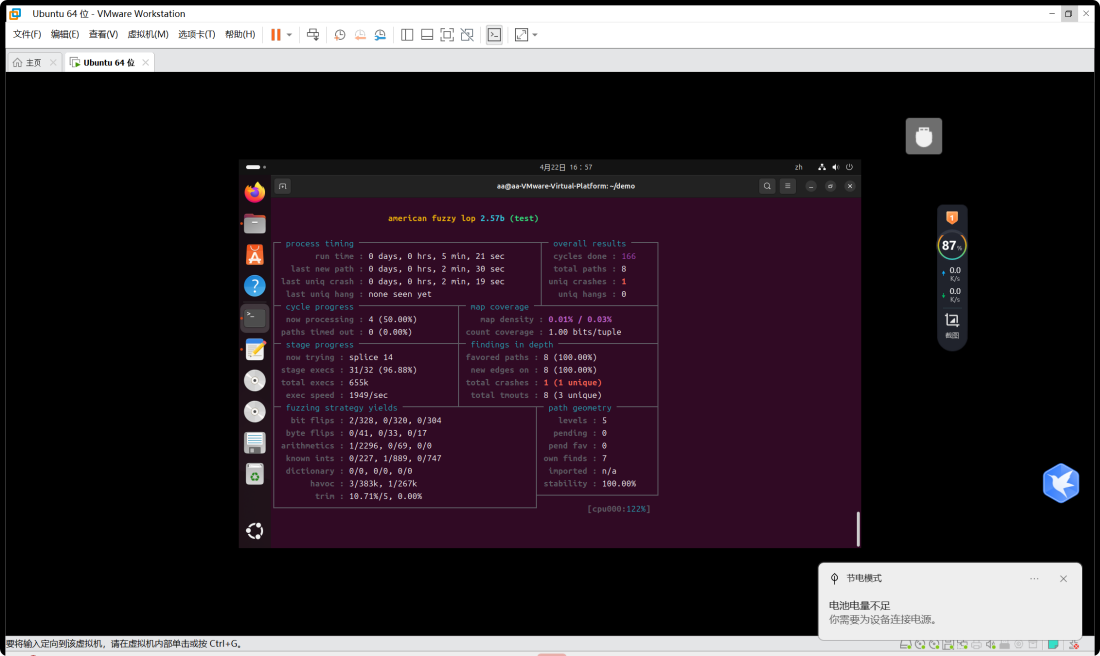
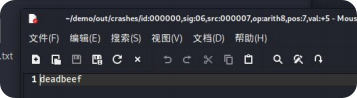


图 6启动模糊测试后可以看到此界面

1. Crash分析



Out的子文件Crash文件夹下可以看到deadbeef，正是使我们代码产生崩溃的输入。

**心得体会：**

通过本次AFL模糊测试实验，我系统掌握了漏洞挖掘的重要工具和方法，对网络安全攻防技术有了更深刻的理解。从Linux虚拟机的搭建、AFL工具的安装配置，到编写测试程序、执行模糊测试，每一个环节都让我体会到理论与实践结合的重要性。在操作过程中，我不仅熟悉了AFL工具的安装和使用流程，更深入理解了覆盖引导和文件变异的核心概念。

在模糊测试中，覆盖引导是一种关键技术。它主要通过监测程序在执行过程中的路径和代码覆盖情况来工作。AFL 等模糊测试工具会在程序中插入一些特殊的代码（称为插桩），这些插桩能够记录程序执行到了哪些代码块、哪些分支被执行了等信息。通过分析这些覆盖信息，模糊测试工具可以了解到哪些部分的代码还没有被充分测试到，然后针对性地生成新的测试用例，以尝试覆盖那些未被覆盖的代码区域。这样可以提高测试的全面性和有效性，增加发现潜在漏洞的可能性。这种基于动态分析的测试方式，让我意识到漏洞挖掘的高效性和创新性。