《漏洞利用及渗透测试基础》实验报告

姓名： 许健 学号： 2013018 班级： 信安班

**实验名称：**

AFL模糊测试实验

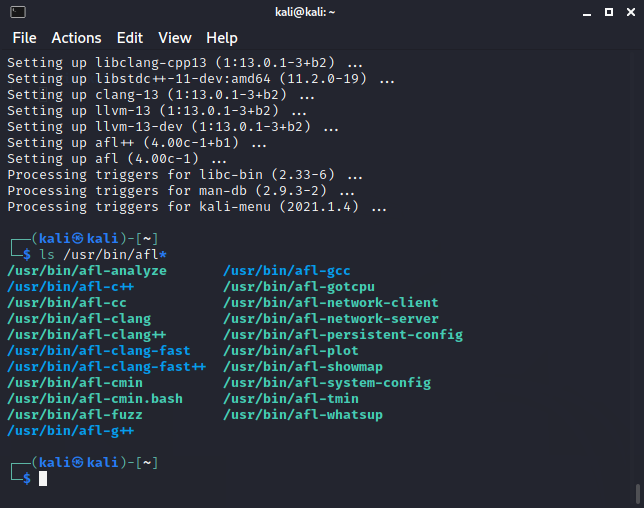
**实验要求：**

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：**

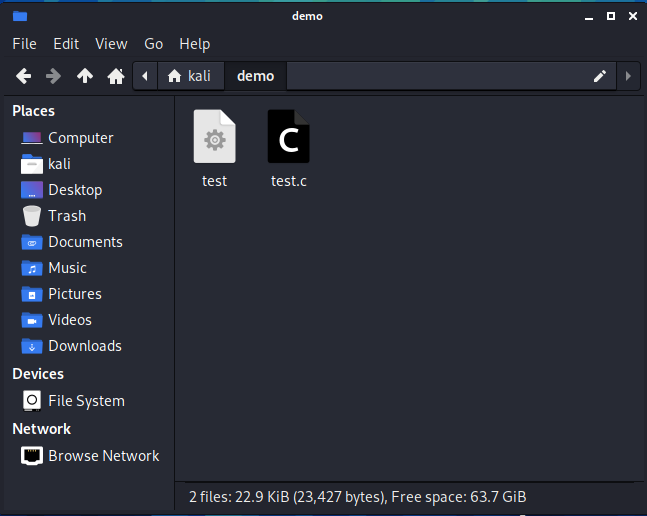
1. **AFL安装**

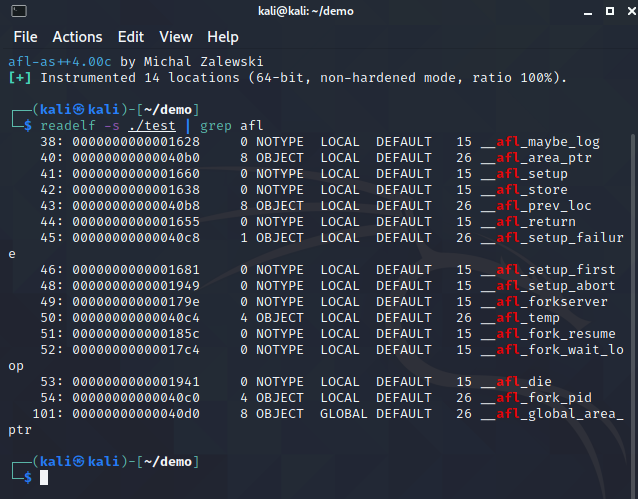
查看路径可以看到afl安装的文件：ls /usr/bin/afl\*，如图



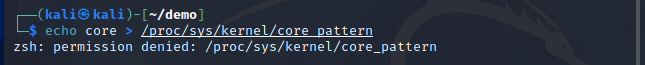
1. **AFL测试**

创建本次实验的程序并编译，命令：**afl-gcc -o test test.c**

查看编译后的插桩：



输入命令指示系统将coredumps输出为文件，而不是将它们发送到特定的崩溃处理程序应用程序。（之前已经做过）



命令：**echo core > /proc/sys/kernel/core\_pattern**

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。

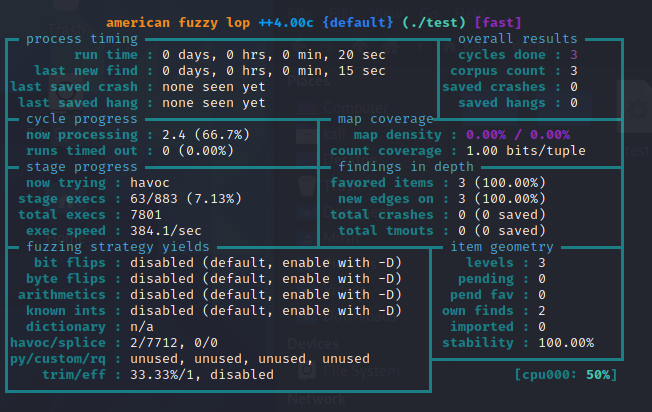
然后，在输入文件夹中创建一个包含字符串“hello”的文件。

foo就是我们的测试用例，里面包含初步字符串hello。AFL会通过这个语料进行变异，构造更多的测试用例。

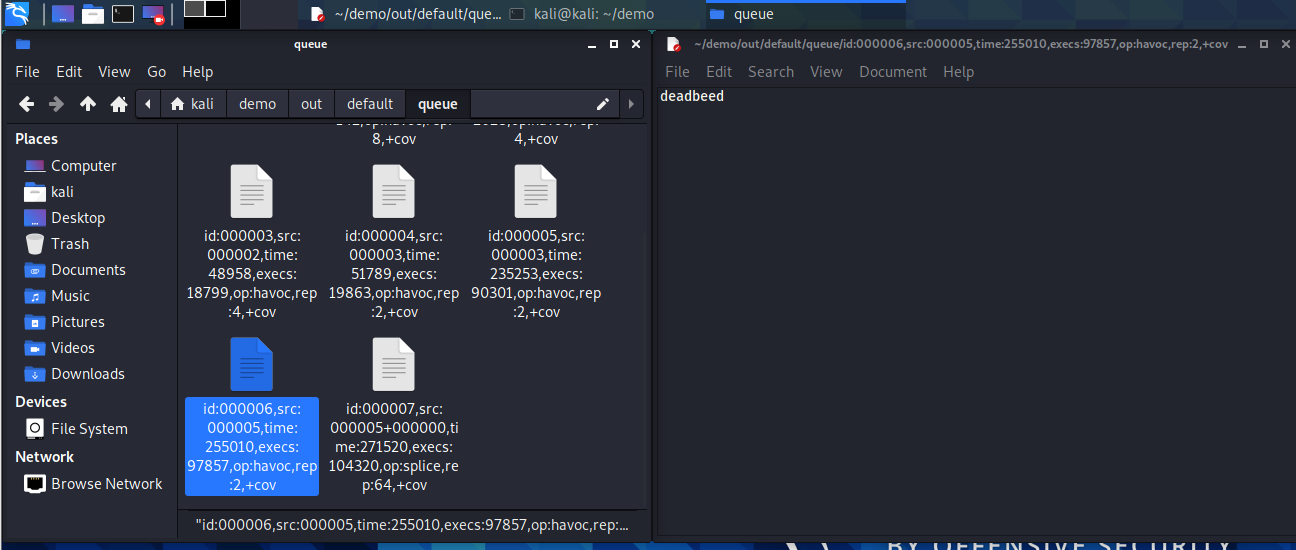


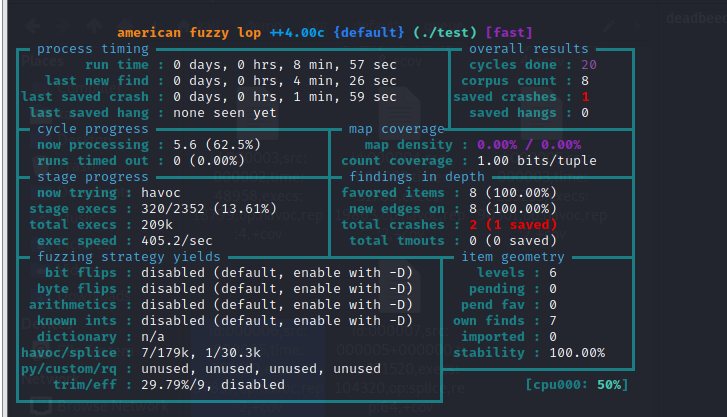
1. **启动模糊测试**

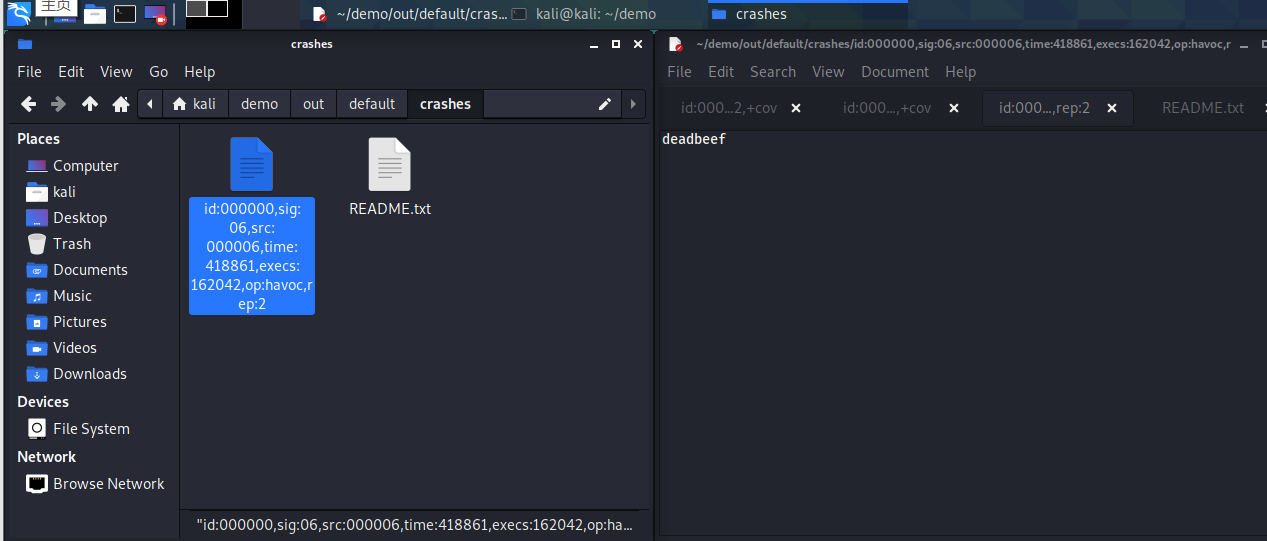
运行界面：



执行过程中查看队列中的文件，不断接近“deadbeef”：



找到crash,可以看到运行的时间以及观察fuzzing结果，如有crash，定位问题



在out文件夹下的crashes子文件夹里面是我们产生crash的样例

hangs里面是产生超时的样例

queue里面是每个不同执行路径的测试用例。

通常，得到crash样例后，可以将这些样例作为目标测试程序的输入，重新触发异常并跟踪运行状态，进行分析、定位程序出错的原因或确认存在的漏洞类型。

**心得体会：**

模糊测试（Fuzzing）技术作为漏洞挖掘最有效的手段之一

本次实验熟悉了简单的AFL框架的使用，如何用来发现漏洞

概念含义的理解：

覆盖引导：通过记录输入样本的 代码覆盖率，从而调整输入样本以提高覆盖率，增加发现漏洞的概率

文件变异：通过一些特殊的策略对文件的内容进行变换，生成新的测试文件，目的是覆盖更多的执行路径。