**实验二**

**passwd实现细粒度访问控制及root能力安全使用**

**班级：1703202**

**学号：1170300421**

**姓名：贺宗磊**

**第一部分：**

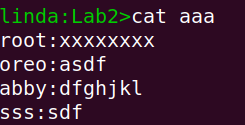
**一、分析passwd程序实现过程，模拟系统中密码修改机制，在自主访问控制系统中实现细粒度的权限管理**

**实验要求：**

自己编制文件和程序，仿制passwd程序修改/etc/shadow的功能，包括：

a)自己设置一个类/etc/shadow文件aaa，该文件中约定好内容格式，和读取该文件的程序相配合，文件中包括超级用户及其内容、普通用户及其内容

文件设置如下：



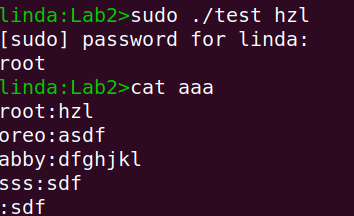
b)编制程序使得：Root用户能够读取和修改aaa文件中所有用户的内容普通用户仅能够读取和修改aaa文件中属于自己用户的内容

程序如下：

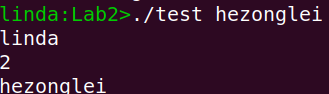


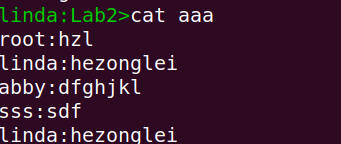


以root身份运行修改root名：



以普通身份修改用户名：

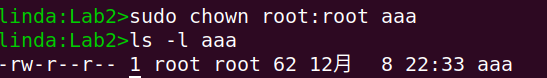




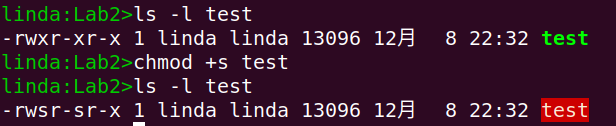
c）普通用户能以root身份执行所编制的类passwd程序

将aaa文件拥有者改为root:





给可执行文件test设置suid位：



**第二部分：**

二、**root 的capability使用**

实验内容：

1.修改系统内核，配置capability的能力位，实现几种能力位的设置可验证。

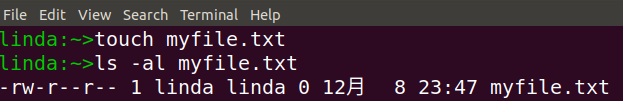
实验过程：

（1）实现３种基本能力位的授权和查看，并分析授权前和授权后的差异。

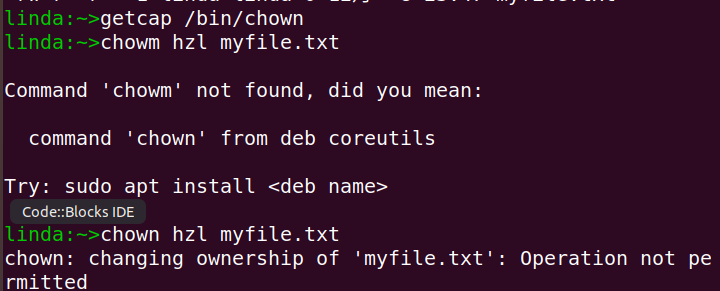
1.Cap\_chown

该能力位，设置后，可以修改文件属主的权限。

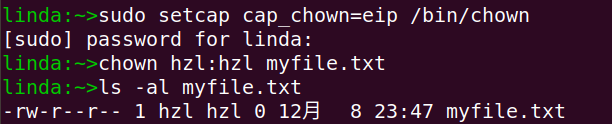
首先创建一个属于用户的文件，这里就创建了一个属于 linda 的文本文件：



Setcap前，查看当前的能力，尝试修改文件属主：



执行 Setcap 命令,更改文件所有者，由 linda 更改为 hzl ：



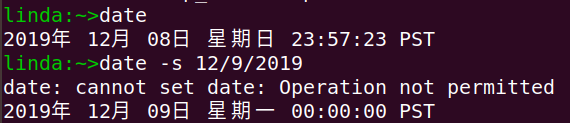
查看 cap ：



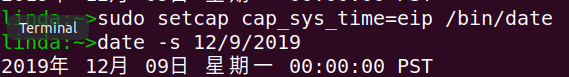
2.cap\_sys\_time

该能力位，设置后，允许修改系统时钟

Setcap前，查看当前时间，并尝试修改日期，修改失败：



Setcap：Setcap 后，修改日期，修改成功：



查看 cap ：

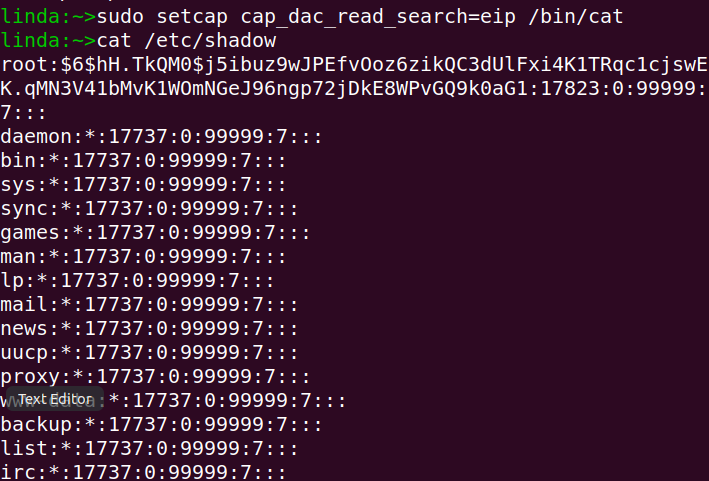


3.cap\_dac\_read\_search，忽略所有对读、搜索操作的限制：

Setcap 前，尝试读 etc/shadow 文件：



Setcap 后，可以查看了：



（2）系统启动时关闭某能力位，对系统的应用和安全性有何影响，以具体能力位为例说明，比如cap\_sys\_module，cap\_linux\_immutable

如果关闭了 cap\_chown 能力位，那么所有的程序或用户都可以修改文件属主，那么就相当于可以访问或执行任何文件。

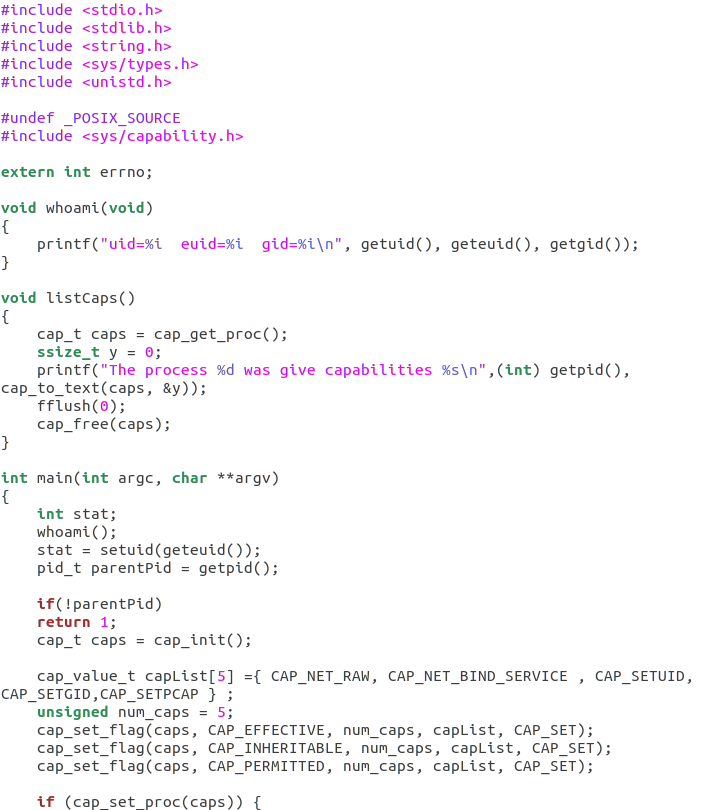
如果关闭了 cap\_sys\_time 能力位，那么所有的程序或用户都可以修改系统的时间，会造成系统的时间错乱。

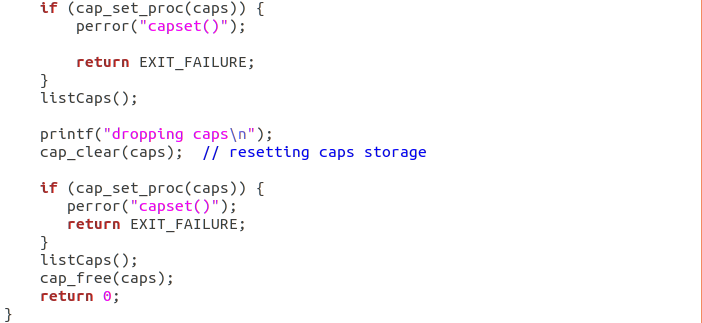
如果关闭了 cap\_dac\_read\_search 能力位，所有的文件都可以被所有的用户读取，这对于一些系统文件来说，是非常不安全的。

（3）组合系统的部分能力位，实现系统的网络管理功能，或用户管理功能、文件管理功能。

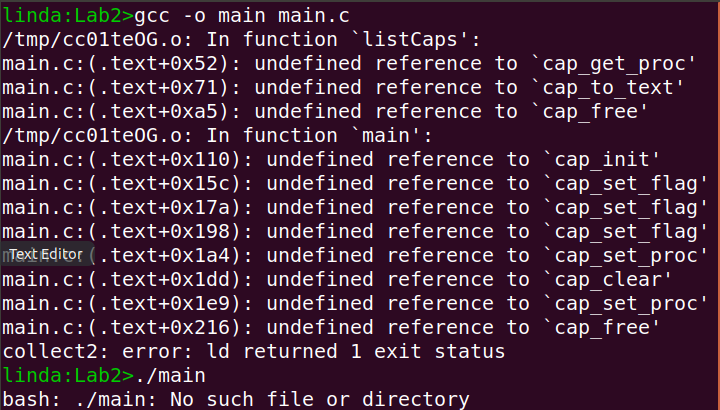
实验设计为给一个特定进程部分能力使其具有相关能力位，并打印出来，再清除掉相关能力，实验设计的是完成文件管理能力。

程序如下：

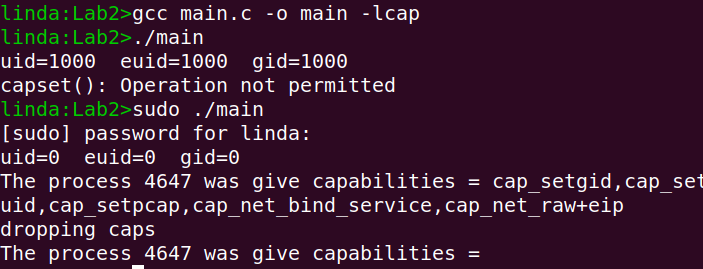




首先先测试一下，没有能力位的时候:

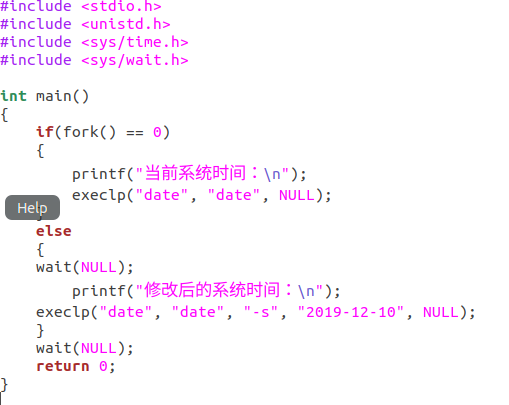


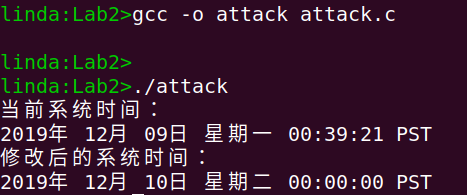
给能力为后（给当前程序进程设定能力，最后清除设定的能力）：



（4）编制攻击程序，测试能力位的安全性。

编写修改系统时间的攻击：





总结：

在本次实验中，首先学习并了解了passwd程序功能，并自己设计了相关程序进行了模拟，进一步加深了了解，之后学习并了解了能力位相关概念，并通过实验进一步加深了了解，在实验中程序运行中曾遇到fatal error: sys/capability.h: No such file or directory，后查询为缺少安装部分文件，利用sudo apt-get install libcap-devan 安装后，程序正常，最终完成实验。