

**计算机系统安全实验报告**

**实验一：文件权限管理及搭建虚拟环境**

学　　 号 1190201308

班　　 级 1903202

学 生 陈东鑫

日 期 2021年11月24日

## 1.1 Linux系统文件权限设置与辨识setuid程序uid差异

#### 设计并实现不同用户对不同类文件的r、w、x权限:

（1）查看系统文件的权限设置

a)查看/etc/passwd文件和/bin/passwd文件的权限设置，并分析其权限为什么这么设置；

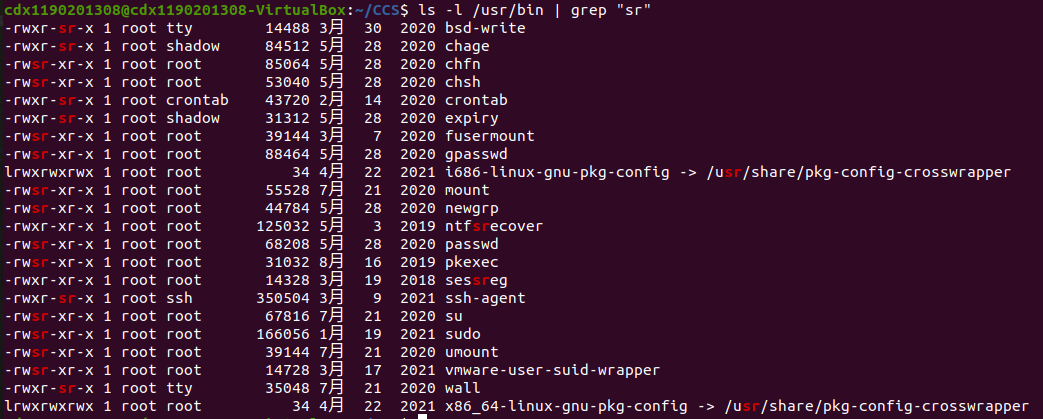
IMG_256

IMG_256

执行命令ls -l [文件名] 查看文件的权限设置，如上图所示。

可以看到两个文件都属于root，但权限不同。对于/etc/passwd，该文件保存了用户信息，只有root用户能进行修改，其他用户只有读权限，保证了系统重要信息的安全性。/bin/passwd是一个二进制可执行程序，用于用户的密码，但密码保存在/etc/shadow文件中，只有root用户才可以进行修改，于是/bin/passwd设置了setuid位，所有用户都可以以root用户身份执行。

b)找到2个设置了setuid位的可执行程序，该程序的功能，该程序如果不设置setuid位是否能够达到相应的功能。



如图所示，查看/bin目录下的可执行程序权限，所有文件拥有者为root、文件拥有者权限为\*\*s且组拥有者或其他用户拥有者有执行权限的文件为设置了setuid位的可执行程序。

以其中两个举例：首先是passwd，由上面的a)介绍，passwd需要用户以root用户身份执行，否则无法打开/etc/shadow文件进行修改密码，即不设置setuid位不能够达到修改密码的功能。

然后是sudo，sudo暂时赋予用户root身份，且需要输入密码确认，如果不设置setuid无法切换身份。

（2）设置文件或目录权限

a)用户A具有文本文件”流星雨.txt”，该用户允许别人下载；

IMG_256

用户css\_test创建用户和组为ccs\_test的文件，将权限设为所有用户可读，则css\_test其他用户可以查看并下载流星雨.txt。

b)用户A编译了一个可执行文件”cal.exe”，该用户想在系统启动时运行；

IMG_256

在系统启动时运行，当时无法确认用户，故需要所有用户的执行权限。

c)用户A有起草了文件”demo.txt”，想让同组的用户帮其修改文件；

IMG_256

ccs\_test加入cdx1190201308用户组，并让同组用户拥有写权限，但其他成员不能有写权限，于是cdx1190201308用户可以修改ccs\_test用户创建的文件demo.txt。

IMG_256

d)一个root用户拥有的网络服务程序”netmonitor.exe”，需要设置setuid位才能完成其功能。

IMG_256

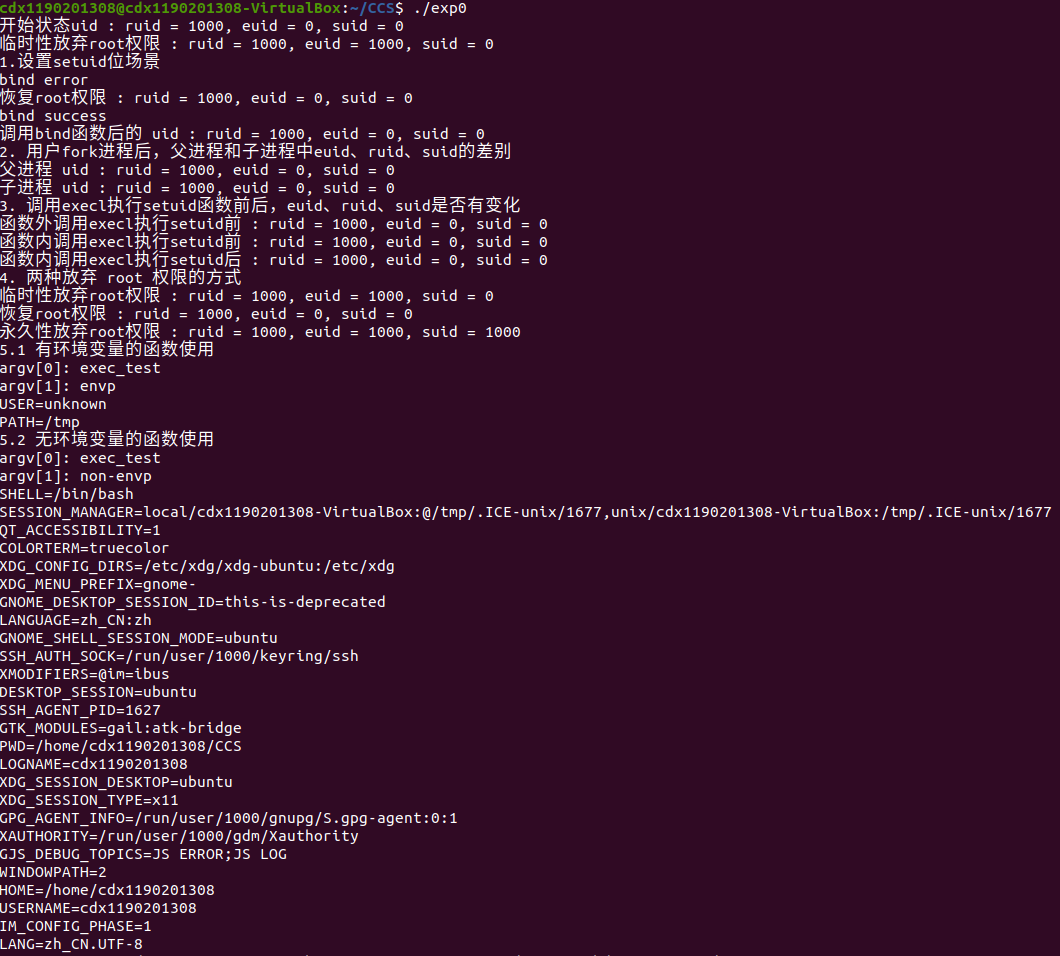
使用命令chmod a+x netmonitor.exe 、chmod u+s netmonitor修改权限，令所有用户拥有执行权限，并设置setuid位。

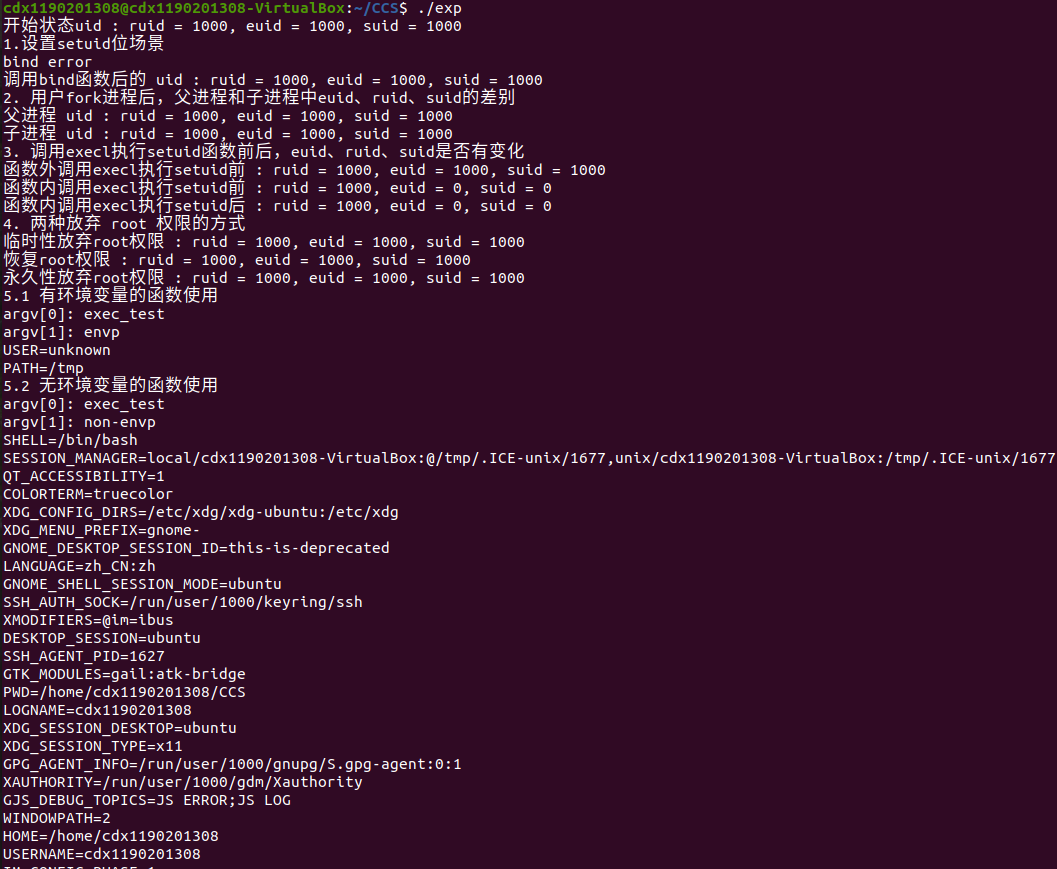
#### 一些可执行程序运行时需要系统管理员权限，在UNIX中可以利用setuid位实现其功能，但setuid了的程序运行过程中拥有了root权限，因此在完成管理操作后需要切换到普通用户的身份执行后续操作：

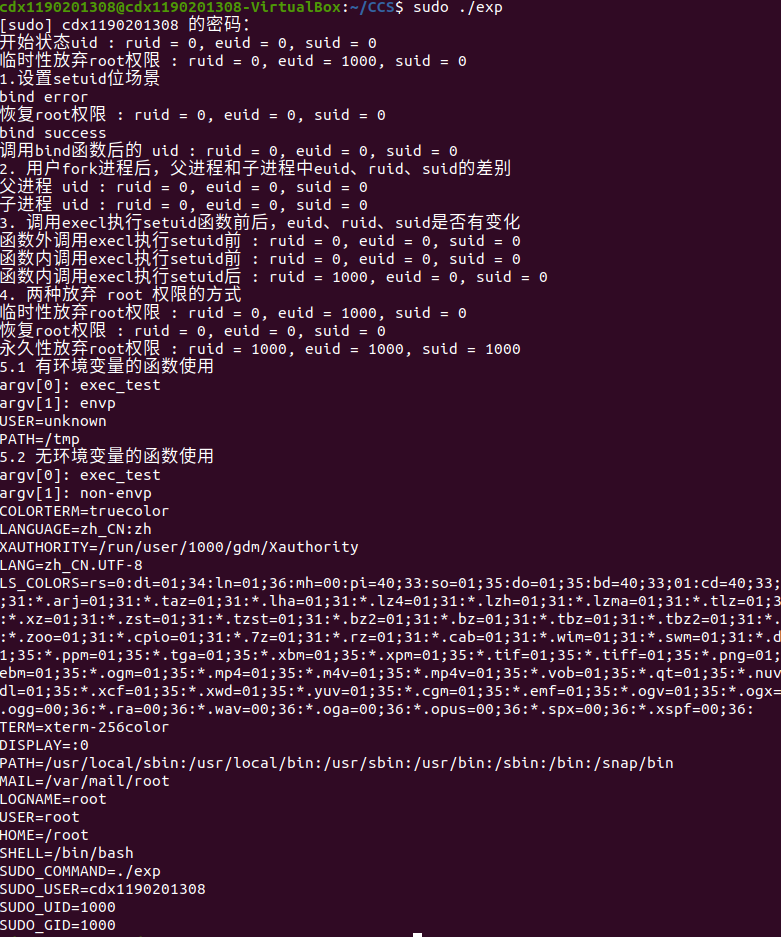
编写程序（见附件），分别为主题程序exp.c和exp0.c，程序内容相同，将exp.c编译为可执行程序exp，将exp0.c编译为exp0并将其修改为setuid位程序，如下图。

IMG_256

程序执行结果如下图（环境变量部分较长，省略部分函数输出的环境变量）







1. 设想一种场景，比如提供http网络服务，需要设置setuid位，并为该场景编制相应的代码；

bind函数绑定小于1024端口号时需要root权限，否则绑定失败，程序中使用了80端口，故需要euid为0才能bind成功。于是./exp0,sudo ./exp能够成功。

1. 如果用户fork进程后，父进程和子进程中euid、ruid、suid的差别；

fork进程后，父进程和子进程ruid、euid、suid不变。

1. 利用execl执行setuid程序后，euid、ruid、suid是否有变化；

execl执行setuid程序后，在进行setuid前，与调用execl前ruid、euid、suid相同。成功调用setuid后，euid和suid变为0。

1. 程序何时需要临时性放弃root权限，何时需要永久性放弃root权限，并在程序中分别实现两种放弃权限方法；

当执行完需要root权限的操作后，若暂时不需要root权限，则暂时放弃root权限，若需要root权限则从suid恢复；若确定程序后面没有需要root权限执行的操作，则永久放弃root权限。

1. execl函数族中有多个函数，比较有环境变量和无环境变量的函数使用的差异。

execle需要调用时输入环境变量，函数将输入的环境变量作为执行程序的环境变量。

execl不需要调用时输入环境变量，程序自动将进程的环境变量作为默认的环境变量。

#### 心得体会。

在本实验该部分学习了文件的类用户权限，学习了使用chmod修改文件的权限，理解了权限的管理方法、变化规律，了解了setuid位的作用。

## 1.2 chroot的配置

#### 利用chroot工具来虚拟化管理

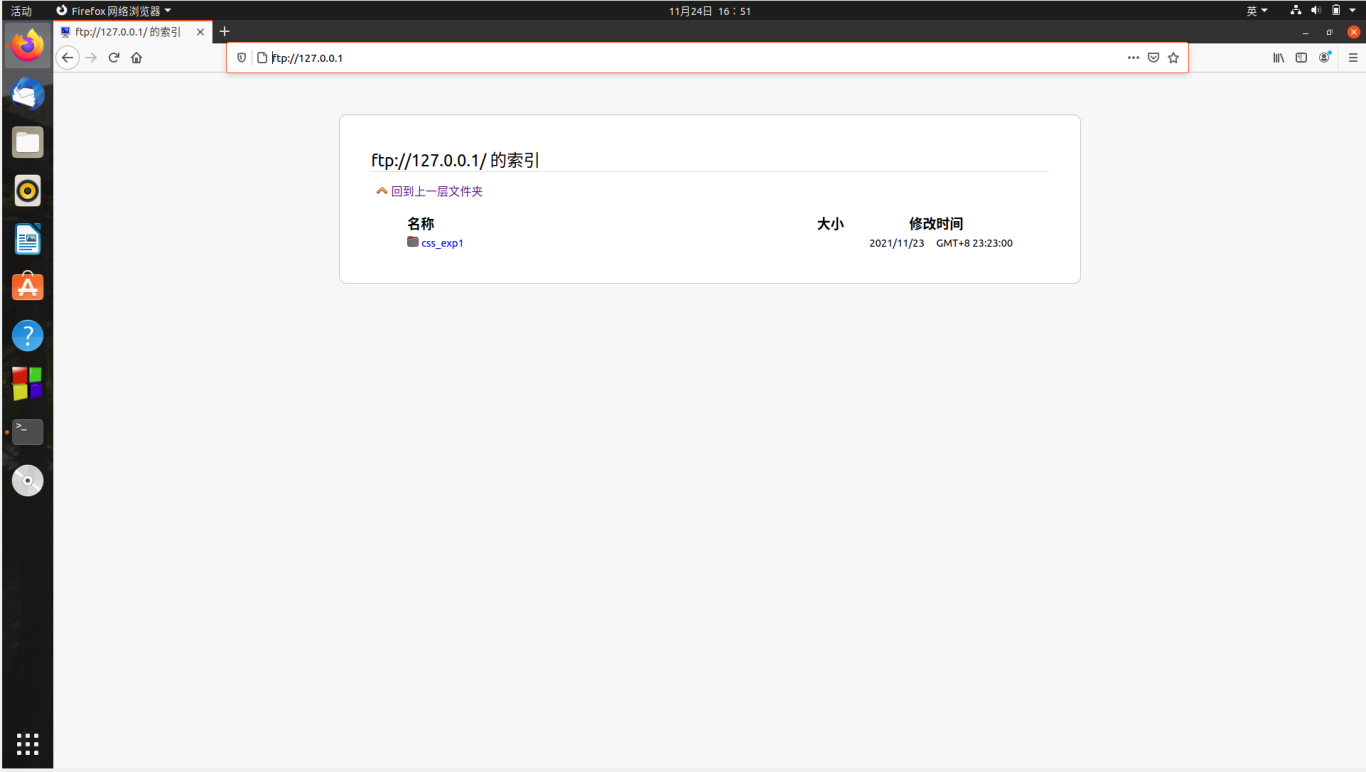
1)实现bash或ps的配置使用；

由于虚拟机中没有找到自带的ftp，于是安装了pure-ftp；在/var/chroot/myftp下建立chroot根目录，使用cp [待复制文件路径] /var/chroot/myftp --parent 将bash和pure-ftp转移到虚拟根目录下，目录结构一同复制；然后使用ldd查看连接文件，继续使用cp命令将文件复制到目录。

运行时使用sudo chroot /var/chroot/myftp/

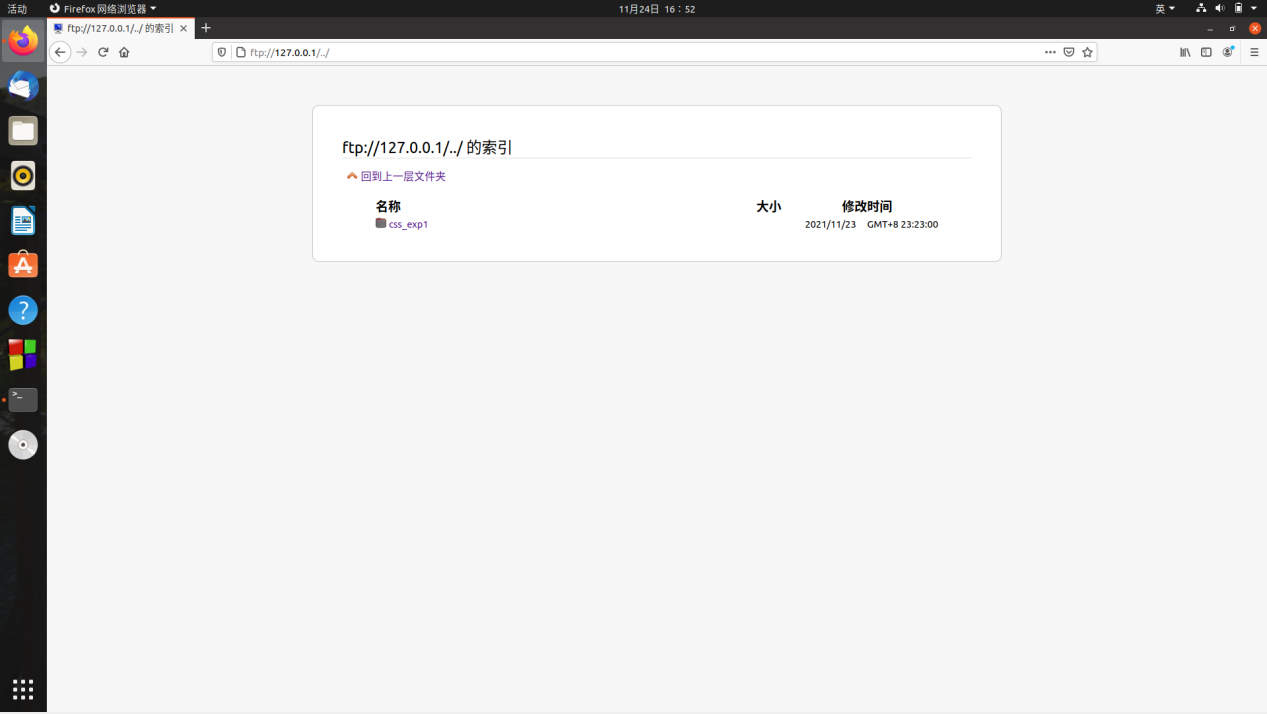
IMG_256

2)利用chroot实现SSH服务或FTP服务的虚拟化隔离；

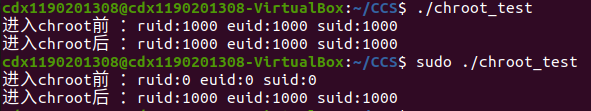


可以在term中使用ftp 127.0.0.1链接本机，建立文件夹和文件等（如上图的ccs\_test文件夹）。

chroot以/var/chroot/myftp作为本目录，点击返回上一级，仍然无法返回上一级。做到了隔离。



3)chroot后如何降低权限，利用实验一中编制的程序检查权限的合理性；



编写程序chroot\_test.c，在程序内调用chroot，查看调用前后权限变化。

可以看到进入chroot后，用户的权限被重置了，即使使用root权限进入也会永久放弃root权限。

4)在chroot之前没有采用cd xx目录，会对系统有何影响，编制程序分析其影响。

执行chroot命令时，后面的目录是作为虚拟根目录的目录，如果该目录写错，虚拟目录则设置为那个目录；若虚拟目录写到上一级或几级，则访问者可以访问本不可以访问上级目录，可能造成泄露。

#### 心得体会。

本实验该部分学习了使用chroot虚拟化限制方法搭建安全的沙盒环境，实现安全的系统加固，在环境中保障安全性，对权限的限制和虚拟化隔离有了更深的理解。