实验一

要求：

（1） 设想一种场景需要进行普通用户和 root 用户切换，设计程序实现 euid 的安全管理

配合第 3 章 完成进程中 euid 的切换，实现 root 权限临时性和永久性管理，加强程序的安全性

说明：1 学时，不分组实现

（2） 搭建安全的沙盒环境，在沙盒环境中提供必须的常见工具，并提供程序验证沙盒环境的安全性

配合第 3 章 实现系统中的虚拟化限制方法，实现安全的系统加固，测试虚拟化空间的加固程度

说明：3 学时，2 人一组，分组实现

1.1 Linux 系统文件和目录权限设置与辨识 setuid 程序 uid 差异

1、 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/etc/bin/passwd 文件的权限设置，并分析其权限为什么这么设置；

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，

（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；

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；

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；

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。

2、 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。

(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；

(2)如果用户 fork 进程后，父进程和子进程中 euid、ruid、suid 的差别；

(3)利用 execl 执行 setuid 程序后，euid、ruid、suid 是否有变化；

(4)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；

(5)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。

3、 编制实验报告，对问题一说明原理，对问题 2 说明设计过程和实验步骤。并写出心得体会。

1.2 chroot 的配置

1、 利用 chroot 工具来虚拟化管理

1) 实现 bash 或 ps 的配置使用；

2)利用 chroot 实现 SSH 服务或 FTP 服务的虚拟化隔离；

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；

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。

可参考后面的文件内容进行处理。

2、 编制实验报告，说明原理，设计过程和实验步骤。并写出心得体会。

Ubuntu 下 chroot FTP 服务

1、准备基本的 chroot 环境

在进入 chroot 环境之前要先准备好相应的设置，在本例中是将 ftpd chroot 到/var/chroot 目录中。因为系统自带的 ftpd 在/usr/libexec/目录，所以需要在/var/chroot 中执行以下操作：

#mkdir -p /var/chroot/usr/libexec

然后将 ftpd 复制到该目录中：

#install -C /usr/libexec/ftpd /var/chroot/usr/libexec

将 ftpd 需要的库也复制到 chroot 目录中，使用 ldd 来检测 ftpd 运行时需要哪些库文件：

ldd /usr/libexec/ftpd

/usr/libexec/ftpd:

libskey.so.2 => /usr/lib/libskey.so.2 (0x28074000)

libmd.so.2 => /usr/lib/libmd.so.2 (0x2807b000)

……

ldd 的运行结果显示了 ftpd 运行时需要库，把这些库安装到 chroot 的相应目录中：

mkdir -p /var/chroot/usr/lib

install -C /usr/lib/libskey.so.2 /var/chroot/usr/lib

install -C /usr/lib/libmd.so.2 /var/chroot/usr/lib

……

2、配置 chroot 环境

2.1 检查 ftpd 是否能在 chroot 环境中运行：

chroot /var/chroot /usr/libexec/ftpd

ELF interpreter /usr/libexec/ld-elf.so.1 not found

程序出错，根据提示在/usr/libexec 中还缺少文件 ld-elf.so.1，由于 ftpd 是在 chroot 环境中运行，所以应将 ld-elf.so.1 复制到 chroot 环境中，即/var/chroot/usr/libexec 中：

install -C /usr/libexec/ld-elf.so.1 /var/chroot/usr/libexec

2.2 再次尝试进入 chroot 环境：

chroot /var/chroot /usr/libexec/ftp

这次没有任何提示说明运行库已经准备好了，但是由于 ftpd 在不带-D 参数的时候运行完后就会自动退出，所以现在还是无法从远程登录 ftp 服务，试着在 ftpd 后面加上参数-D：

chroot /var/chroot /usr/libexec/ftpd –D

结果与上次一样，通过查阅 chroot(8)的手册，可以看到 chroot 的语法是：

chroot newroot [command]

也就是说 command 后面不能带参数，即然这样可以写一个简单的 shell 脚本来运行 ftpd，脚本命名为 ftpd.sh，存放于/var/chroot/usr/libexec 中，内容为：

#!/bin/sh

#/usr/libexec/ftpd -D -4

由于不需要支持 IPv6，所以这里加上了参数-4 只对 IPv4 提供支持，当然也可以加上一些其它参数。接下来为脚本加上执行权限：

#chmod a+x /var/chroot/usr/libexec/ftpd.sh

为了要运行这个脚本程序，还需要将/bin/sh 到的 chroot 环境中：

#mkdir /var/chroot/bin

#install -C /bin/sh /var/chroot/bin

接下来就要为 chroot 环境准备/etc 目录了。首先要复制的就是/etc/services 文件，因为它定义了 ftpd 使用的端口号和协议：

mkdir /var/chroot/etc

cp /etc/services /var/chroot/etc

因为需要验证用户，所以需要复制 master.passwd 和 group：

cp /etc/group /var/chroot/etc

cp /etc/master.passwd /var/chroot/etc

编辑/var/chroot/etc/master.passwd 和/var/chroot/etc/group，删除不需要使用 ftp 的用户和不必要的组，注意，当更改了 master.passwd 后一定要使用 pwd_mkdb 来生成密码数据库，由于此时需要将密码数据库文件存放在/var/chroot/etc 中，而不是默认/etc 中，所以在 pwd_mkdb 后面加上-d 参数来指定数据库存放位置：

#pwd_mkdb -d /var/chroot/etc /var/chroot/etc/master.passwd

此时如果执行成功的话将会在/var/chroot/etc/目录中多“pwd.db、spwd.db”两个文件。

2.3 再次进入 chroot 环境：

#chroot /var/chroot /usr/libexec/ftpd.sh

现在便可以登录到 chroot 环境下的 ftp 服务器了。

3、结尾工作

为每一个用户建立 home 目录，注意是在建在/var/chroot/home 之中。在/var/chroot/etc/中生成 ftpusers 文件，将禁止登录 ftp 的用户的用户名加入其中，以禁止部分用户登录。

在/var/chroot/etc/中生成 ftpchroot 文件，它的作用是限制用户只能访问自己的 home 目录中的文件，而不能访问 home 外的任何内容。将要限制的用户用户名加入其中。

在/var/chroot/etc/中生成 ftpwelcome 文件，它的作用是当用户连接上服务器的时候显示欢迎信息。在/var/chroot/etc/中生成 ftpmotd 文件，它的作用是当用户登录进服务器的时候显示欢迎信息。