****

**实验报告**

**课程名称： 计算机系统安全**

**实验名称： 文件权限管理及搭建虚拟环境**

**实验日期： 2018.12.5**

**班 级： 1603102**

**姓 名： 安宏展**

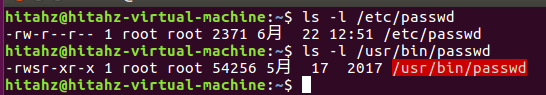
**指导教师： 张玥**

**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；



使用ls -l指令查看各种用户的权限，拥有相应的rwx权限就会在下面显示出来，没有显示-

A.所有的用户在本机工作的时候都要在/etc/passwd下面有一条记录，并且他们可以知道主机有其他的用户，但是root可以修改这些用户的信息

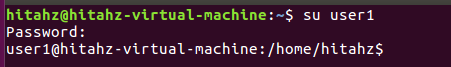
B.组用户和普通用户的执行权限通过root获得的

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

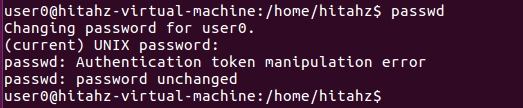


所有用户登陆输入密码都需要这个文件夹的权限

当普通用户登陆时：可以随意修改密码



当删除这个权限的时候：我们想要修改密码的时候就会出现问题



所以不设置setuid位不能够达到相应的功能

同理还有：



1. 设置文件或目录权限

利用chmod xxx document指令设置文件权限，其中rwx权限分别对用4、2、1，设置这种权限就将这类用户的值加上相应的数，比如 chmod 777代表所有用户都拥有对该文件的rwx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；



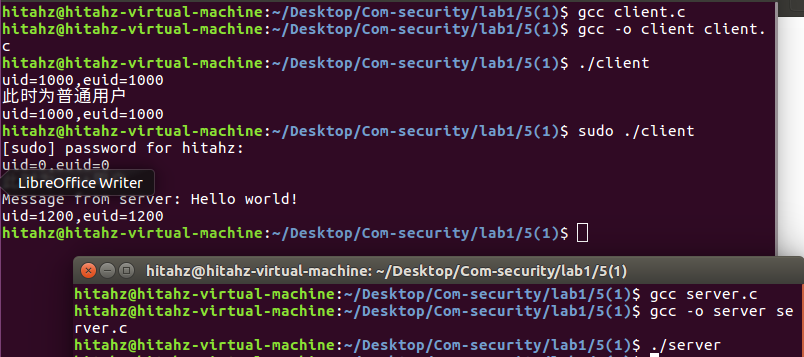
c)用户A有起草了文件”demo.txt”，想让同组的用户帮其修改文件



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。

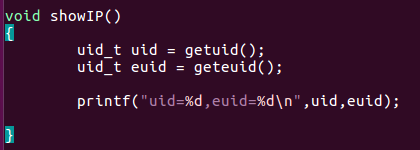


1. 一些可执行程序运行时需要系统管理员权限，在UNIX中可以利用setuid位实现其功能，但setuid了的程序运行过程中拥有了root权限，因此在完成管理操作后需要切换到普通用户的身份执行后续操作。
2. 设想一种场景，比如提供http网络服务，需要设置setuid位，并为该场景编制相应的代码；

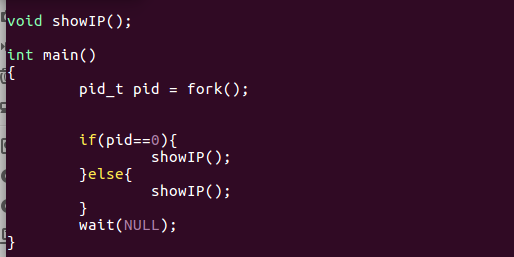


分别使用普通用户和root用户执行client文件，server文件保持不变，会发现普通用户并没有数据传输，而root用户界面中有这样一段，说明root用户的时候进行了http网络服务，普通用户没有进行改服务。

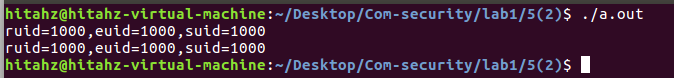
展示uid和euid的函数showIP()



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



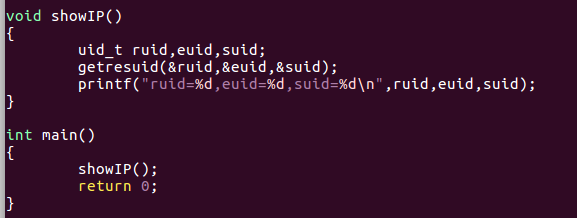
使用一个fork()函数，每个进程内部都执行一个showIP函数，函数同上，执行结果为：



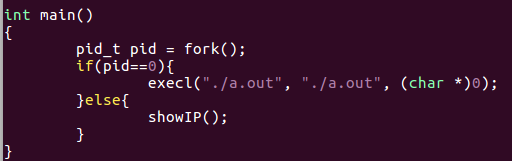
说明用户进入fork进程后，父进程和子进程中三种uid都是不变的

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

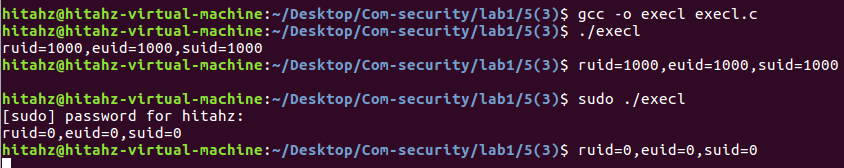
首先编写test.c用于打印三种uid



然后编写带有execl函数的文件execl.c

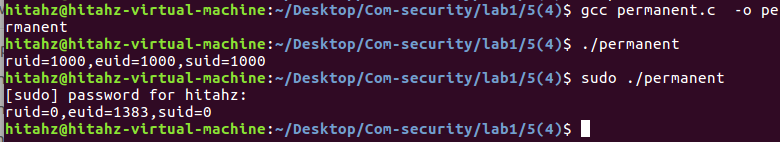


由于执行过execl函数之后就会强制程序停止，所以让execl函数里面执行的函数打印三种uid即可，将前面的test程序编译成a.out，执行结果如下：

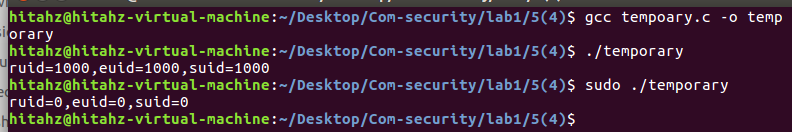


1. 程序何时需要临时性放弃root权限，何时需要永久性放弃root权限，并在程序中分别实现两种放弃权限方法；

永久性放弃权限：



暂时性放弃权限：



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。

查找方式：以’p’结尾的函数只需在参数中给出文件名，然后函数会自动在环境变量“$PATH” 所指定的路径进行查找

另外四种函数需要在参数中给出文件名以及路径才行

参数传递方式：exec函数族的参数传递有两种：一种是逐个列举的方式，而另一种则是将所有参数整体构造指针数组传递。在这里是以函数名的第5位字母来区分的，字母为 "l"(list)的表示逐个列举参数的方式，其语法为const char \*arg;字母为“v”(vector)的表示将所有参数整体构造指针数组传递，其语法为 char \*const argv[]。这里的参数实际上就是用户在使用这个可执行文件时所需的全部命令选项字符串（包括该可执行程序命令本身）。要注意的是，这些参数必须以NULL结束。

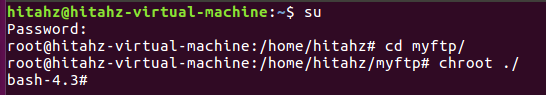
环境变量： exec函数族可以默认系统的环境变量，也可以传入指定的环境变量。这里以 “e”(environment)结尾的两个函数 execle()和 execve()就可以在 envp[]中指定当前进程所使用的环境变量。

心得体会：动手去做，不仅长见识，而且加深了对可能知识的理解，比如放弃权限那部分，如果不动手就永远都不会发现不能使用setuid()函数。所以以后要多动手，尤其是系统方面的内容，内容多且复杂，必须要付诸实践。

* 1. **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；

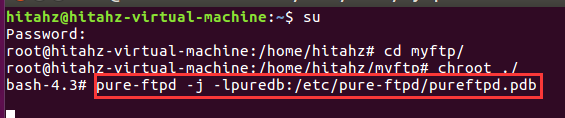


2)利用chroot实现SSH服务或FTP服务的虚拟化隔离；

创建文件夹myftp

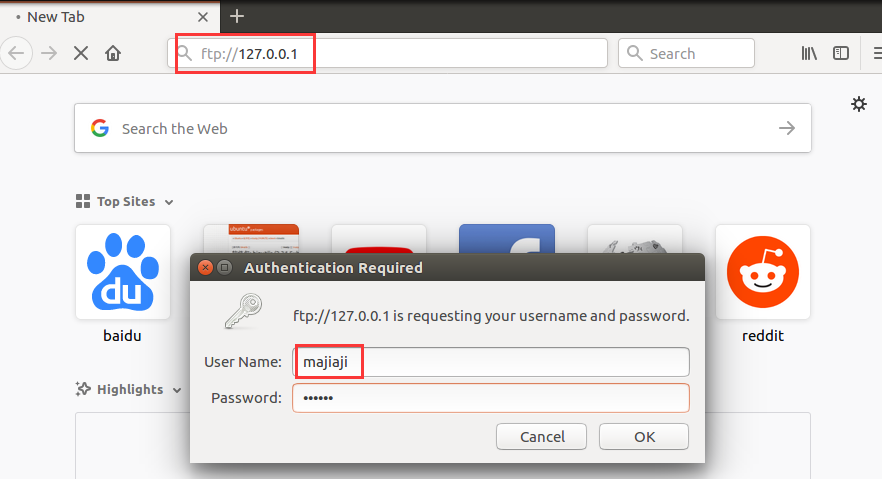
创建新用户majiaji,将其加入预先创建的ftpgroup组里面

在bash中打开连接127.0.0.1，使用浏览器在本地接收

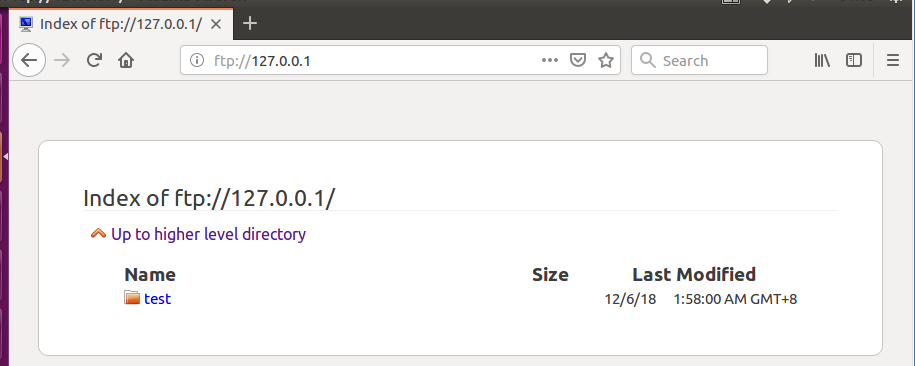


浏览器显示：

首先需要登陆，使用用户名majiaji

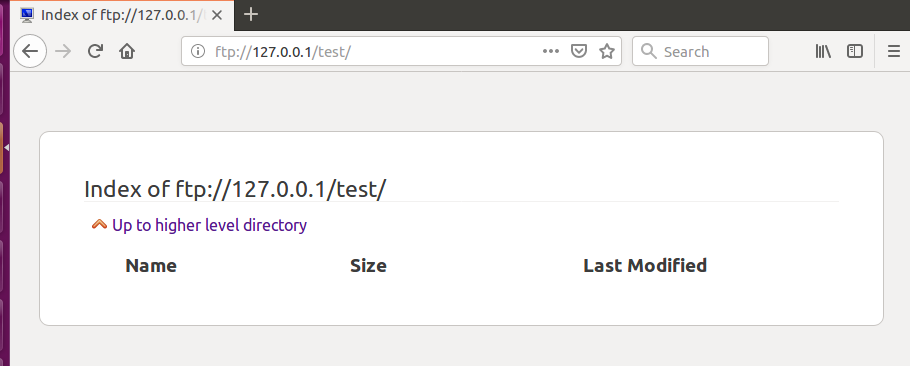


登陆后会发现之前在myftp里面创建的test文件夹

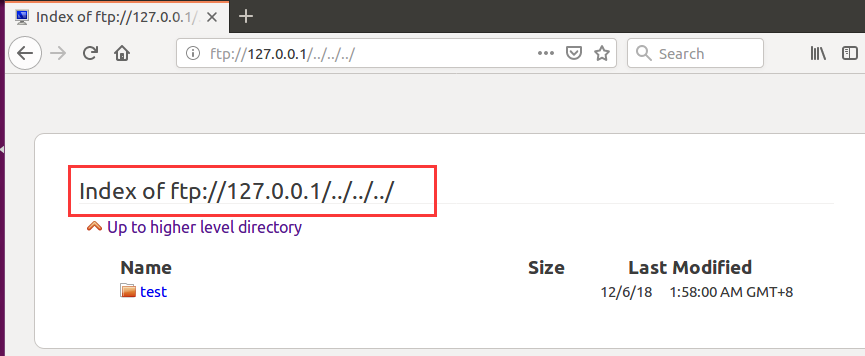


虚拟化隔离：会发现每次想进入test的上一级目录都会失败，但是可以进入test的下一级目录：

test的下一级



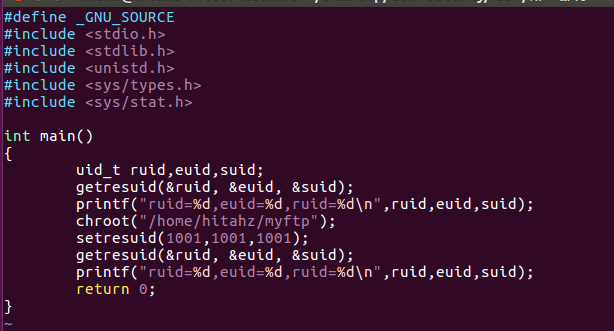
test上一级



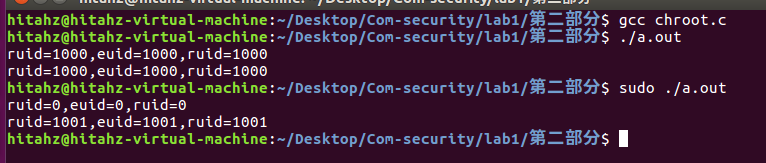
3)chroot后如何降低权限，利用实验一中编制的程序检查权限的合理性；

chroot在root权限下后会将uid设置成非root的uid，但是在非root权限下并不会修改uid

代码如下：

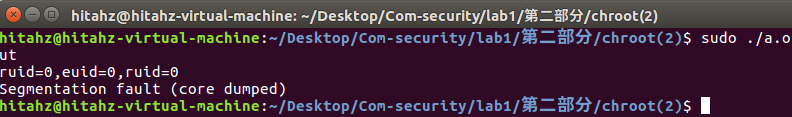


执行结果如下：

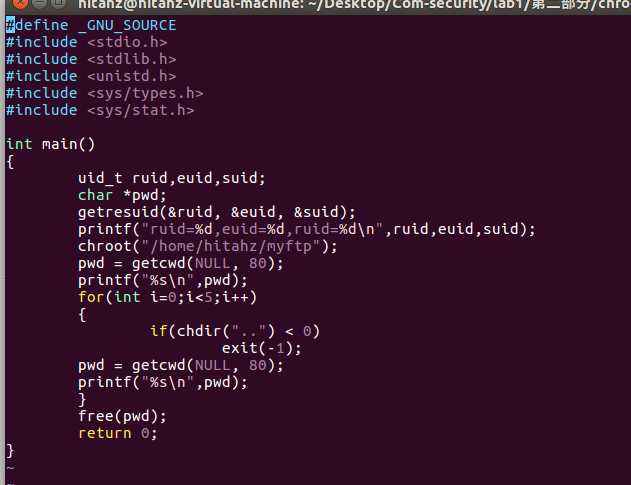


4)在chroot之前没有采用cd xx目录，会对系统有何影响，编制程序分析其影响。

可能会访问不能访问的内存，造成核心转储的问题



代码如下：



心得体会：加深了对chroot以及监狱的理解，操作系统真的很复杂，规则非常多，但是正因为如此多的规则才使得我们的电脑运行的如此流畅，更安全。