**计算机系统安全实验报告**

**实验名称：passwd实现细粒度访问控制及root能力安全使用**

**班级：1703101**

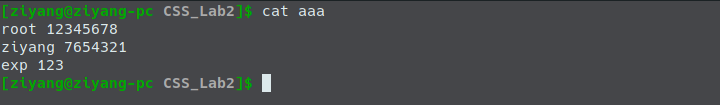
**学号：1170300520**

**姓名：郭子阳**

计算机学院

1 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。

类/etc/shadow文件aaa格式设置为，用户名+空格+密码，如下：



编写passwd.c文件模拟/usr/bin/passwd的功能，功能描述如下：

* 所有用户都可使用passwd+密码修改自己的密码
* root用户可使用passwd+用户名+密码修改任意用户的密码

编译后的passwd程序的文件所有者是root，并且设置了setuid位，使任何用户都可以以root身份执行，aaa文件设置了只允许root读写，不允许其他用户读写。

passwd程序首先要获取进程的ruid，以判断执行的用户，由于获取的仅仅只是用户的id，而不是用户的用户名，于是需要调用getpwuid()函数获取对应的用户名：

uid\_t ruid, euid, suid;

struct passwd\* userStruct;

getresuid(&ruid, &euid, &suid);

userStruct = getpwuid(ruid);

并根据argc，即参数的个数来判断，如果参数只有一个（argc == 2），说明只是修改该用户自己的密码，则无需进行权限判断；如果参数有两个（argc == 3），说明调用者想修改其他用户的密码，此时需要判断调用用户是否是root，如果是root才允许，否则设置errno位，输出错误信息：

case 2:

changePassword(userStruct->pw\_name, argv[1]);

break;

case 3:

if(strcmp("root", userStruct->pw\_name) == 0)

{

changePassword(argv[1], argv[2]);

} else

{

errno = EPERM;

perror("passwd");

}

break;

修改密码用的函数changePassword()函数，接受两个参数：username和password。

首先使用getline函数按行读取aaa文件，按照空格拆分每一行，将空格前的内容与username相比较，如果相同，则说明需要修改该行：