# 计算机系统安全实验报告

## 实验二　passwd 实现细粒度访问控制及 root 能力位安全应用


**班级**：1903201

**学号**：L190202102

**姓名**：李劲光

**2.1 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。**

```
osboxes@osboxes:~/Desktop/sf_hitics/lab2$ sudo ./a.out testuser 12345
The user name of the ruid is root
change testuser's password to 12345
osboxes@osboxes:~/Desktop/sf_hitics/lab2$ ./a.out testuser 12345
The user name of the ruid is osboxes
passwd: Operation not permitted
osboxes@osboxes:~/Desktop/sf_hitics/lab2$ cat user
testuser 12345
testuser2 testuser1234
osboxes@osboxes:~/Desktop/sf_hitics/lab2$
```

passwd.c 文件用来模拟/usr/bin/passwd，功能是所有用户都可以使用 passwd+password 修改自己的密码，编译后的 passwd 程序的文件所有者是 root，并且设置了 setuid 位，以便任何用户都可以以 root 身份执行。 user.txt 文件设置为只允许 root 读写，不允许其他用户读写。

argc 如果只有 len(argc) == 2，表示只修改了用户自己的密码，不需要判断权限，如果 len(argc) == 3，则表示调用如果要修改其他用户的密码，则需要确定调用用户是否为 root。

changePassword() 找到密码所在行后，从下一行读取到文件末尾，将读取的内容保存为字符串，修改后重新写入，然后将文件指针移到密码所在行的开头，按照 username+ ‘ ’ +password 的格式写入并添加换行符，然后将刚才保存的字符串写入文件，写入后使用 ftruncate( ) 功能删除后续内容，防止修改时间比修改后长，修改不完整。

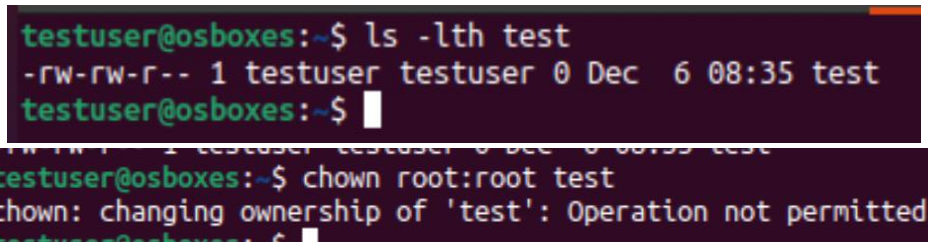
```
testuser@osboxes:/media/sf_hitics/lab2$ ./a.out  123455
The user name of the ruid is testuser
change testuser's password to 123455
testuser@osboxes:/media/sf_hitics/lab2$
```

```
1  osboxes 12345
2  testuser 123455
3  testuser2 testuser1234
4
```

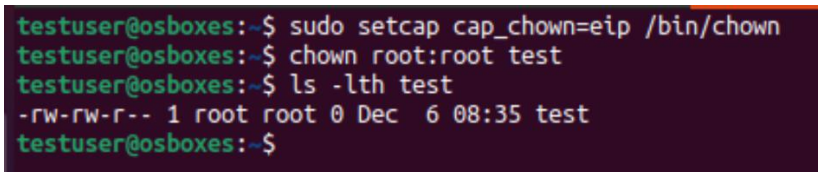
当用户运行修改自己的密码就会提示 Operation not permitted，无法修改。

**2.2 利用 root 的能力机制实现系统加固，有效实现 root 能力的分发和管理。提供程序比较进行 root 能力管理前后系统安全性的差异。**

1) chown 的能力以

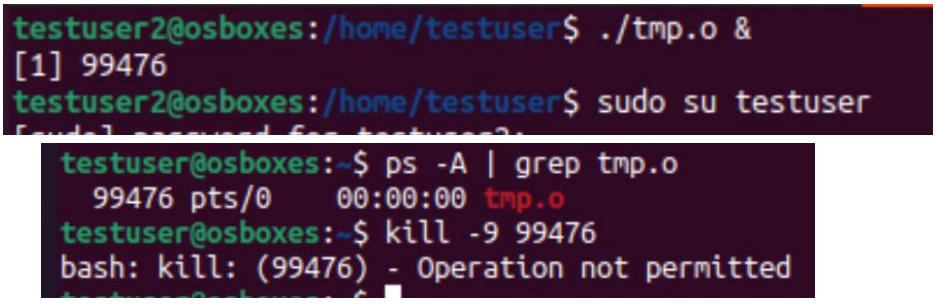



这种能力允许用户任意修改文件的所有者，首先是建立一个属于 testuser 的文件。在修改能力位之前，尝试将文件所有者设置为 root。

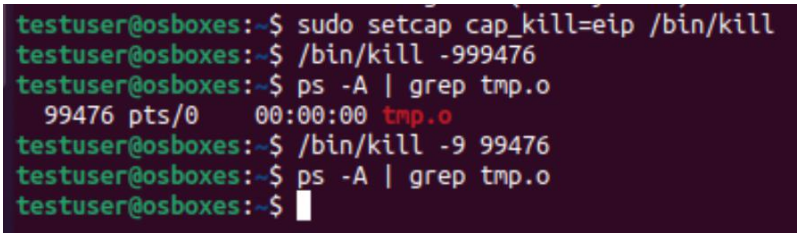



这里执行 sudo setcap 命令之后，再次尝试 chown 修改，即可成功。

2) 如果改变文件名,则能力保留到新文件。



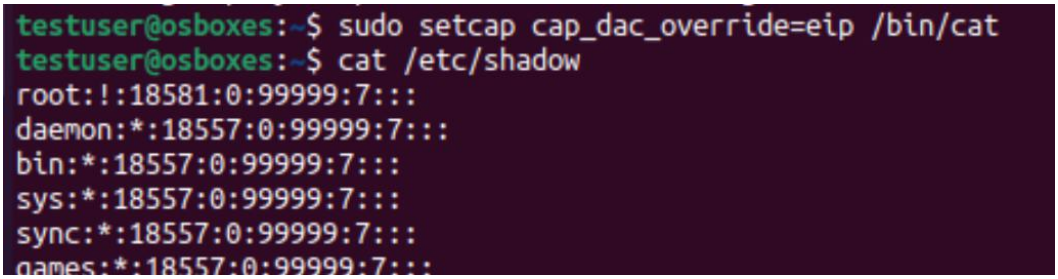

cap_kill 能力位此功能允许用户结束其他用户的执行。 testuser2 在后台执行一个进程。 切换到其他 testuser 时。使用 ps -A 查看进程，然后如果其他用户执行 kill -9 pid 命令会显示 operation not premitted，无法关掉。




我们给 bash 程序分配 cap_kill 发出 kill，就可以关掉其他用户的进程。

3) 用 setcap -r /bin/chown 可以删除掉文件的能力。


cap_dac_override 能力位，无论文件功能位设置是什么，此功能都允许用户读取和写入可执行文件。首先尝试读取/etc/shadow，会显示 permission denied，无法读取。




授权普通用户可以使用 bin/cat 程序修改所有文件的内容，这样就可以读去了/etc/shadow 文件。

4) 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。

如果关闭 cap_chown 位，任何用户都可以随意更改任何文件的用户，相当于可以访问或执行任何文件。如果 cap_kill 位关闭，任何用户都可以关闭任何进程，这可能会关闭系统的重要进程，导致系统不稳定。如果 cap_dac_override 位关闭，则任何用户都可以读取、写入和执行任何文件，而不管权限如何。