****

**实验报告**

**课程名称： 计算机系统安全**

**实验名称： 实验二**

**实验日期： 2018.12.5**

**班 级： 1603102**

**姓 名： 安宏展**

**指导教师： 张玥**

**一、密码修改机制，passwd程序功能的仿制实现**

1、passwd程序功能描述

在Linux中，passwd程序是可信任的，修改存储经过加密的密码的影子密码文件（/etc/shadow），passwd程序执行它自己内部的安全策略，允许普通用户修改属于他们自己的密码，同时允许root修改所有密码。为了执行这个受信任的作业，passwd程序需要有移动和重新创建shadow文件的能力，在标准Linux中，它有这个特权，因为passwd程序可执行文件在执行时被加上了setuid位，它作为root用户（它能访问所有文件）允许，然而，许多程序都可以作为root允许（实际上，所有程序都有可能作为root允许）。这就意味着任何程序（当以root身份运行时）都有可能能够修改shadow文件。

2、实验要求

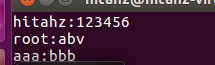
自己编制文件和程序，仿制passwd程序修改/etc/shadow的功能，包括：

1. 自己设置一个类/etc/shadow文件aaa，该文件中约定好内容格式，和读取该文件的程序相配合，文件中包括超级用户及其内容、普通用户及其内容

首先设置好aaa中的内容，分别为：

普通用户:hitahz

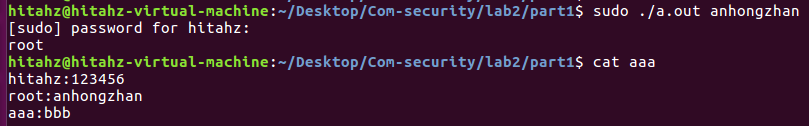
root用户、以及任意用户（或者任意字符串）



1. 编制程序使得：Root用户能够读取和修改aaa文件中所有用户的内容普通用户仅能够读取和修改aaa文件中属于自己用户的内容

使用root权限运行此程序:sudo ./a.out anhongzhan

然后查看aaa中的内容，可以看到root后面的内容已经改为anhongzhan



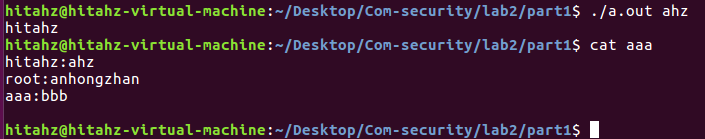
此时可以查看aaa的访问权限，如下图：



可以发现普通用户并没有对于aaa文件的write权限

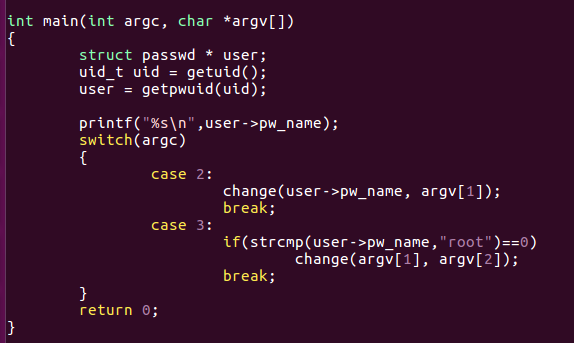
c）普通用户能以root身份执行所编制的类passwd程序

./a.out ahz 运行结果如下图：



注：aaa文件中最后就是存在一个空行，并不是输出错误

代码如下：

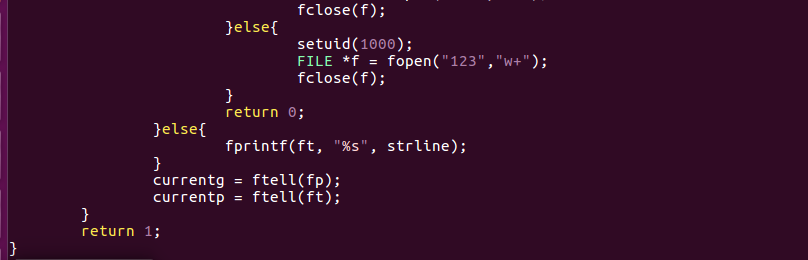


main函数中，通过argc这个参数来判断使用root用户还是普通用户运行程序

argc代表输入参数中字符串的个数，如果用root权限则有sudo，argc=3，这样就可以对两种用户进行区分，然后看change函数







首先寻找root或者是普通用户的名字，然后将名字分号后面的内容替换为text中的内容，其他部分不变。

具体的方法是船建一个文件123，将新的内容写入123中，然后删除文件aaa，将123的改名为aaa

这样写会出现两个问题

第一个是使用root用户创建的文件普通用户没法修改，解决方案就是通过getuid()来判断是否是root用户，如果是则使用setuid()将root用户的权限改变为普通用户。如果是普通用户则无需改动。

第二个问题就是新的aaa对于组用户是没有写权限的，因为正常创建的文件对于组用户拥有写权限，如下图：



但执行程序过后相当于普通用户创建了aaa，故aaa失去了组用户的写权限，如下图：



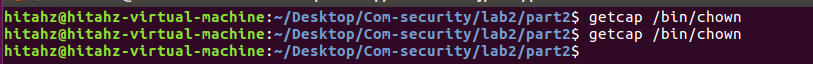
这样写虽然符合要求，但是实际上是不可行的，我们不能改变文件的访问权限，这个问题还需要解决

二、**root 的capability使用**

（1）实现３种基本能力位的授权和查看，并分析授权前和授权后的差异；

cap\_chown:0

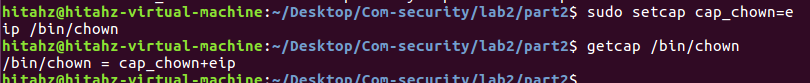
未授予权限时：使用getcap查看不到任何东西



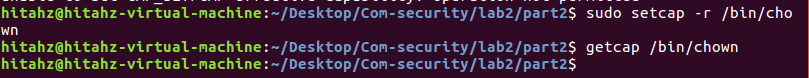
授予权限：

cap\_chown=eip是将chown的能力以cap\_effective(e),cap\_inheritable(i),cap\_permitted(p)

三种位图的方式授权给相关的程序文件



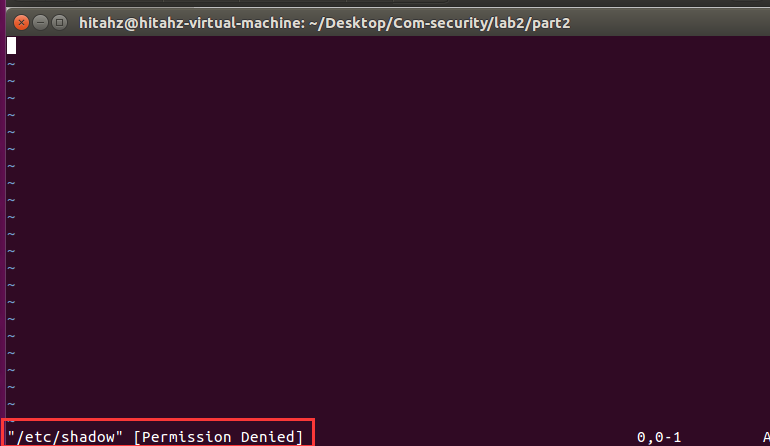
删除权限：



CAP\_DAC\_OVERRIDE : 1

首先在普通用户的情况下查看etc/shadow





会发现无法查看

然后授权CAP\_DAC\_OVERRIDE能力位





CAP\_DAC\_READ\_SEARCH : 2

授权普通用户可以用cat查看任何文件

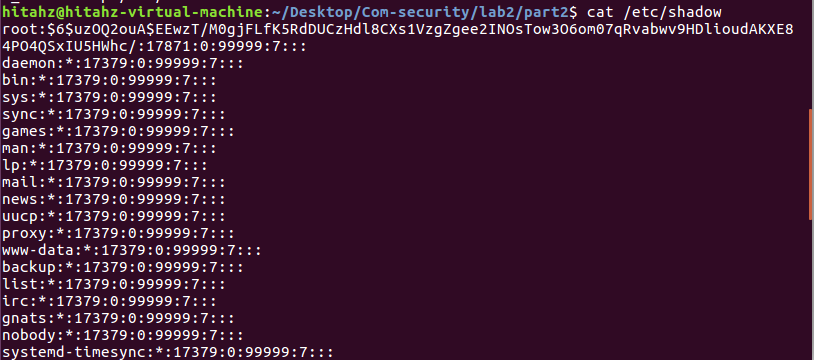
在没有进行授权的情况下：



授权CAP\_DAC\_READ\_SEARCH能力位：



使用cat重新查看



1. 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，以具体能力位为例说明，比如cap\_sys\_module，cap\_linux\_immutable

CAP\_SYS\_MODULE : 16

该能力位允许用户插入和删除内核模块，这是非常危险的，所以关闭该能力位可以提升系统的安全性，但是只有root用户才能修改系统内核，对于系统的应用来说起到了反面作用

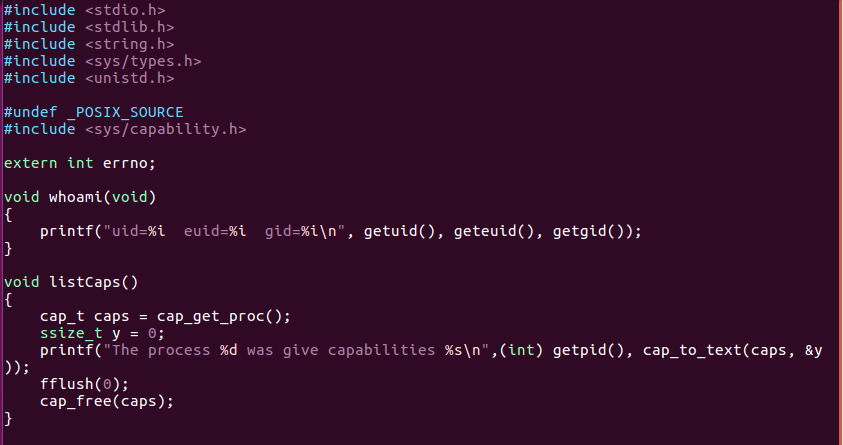
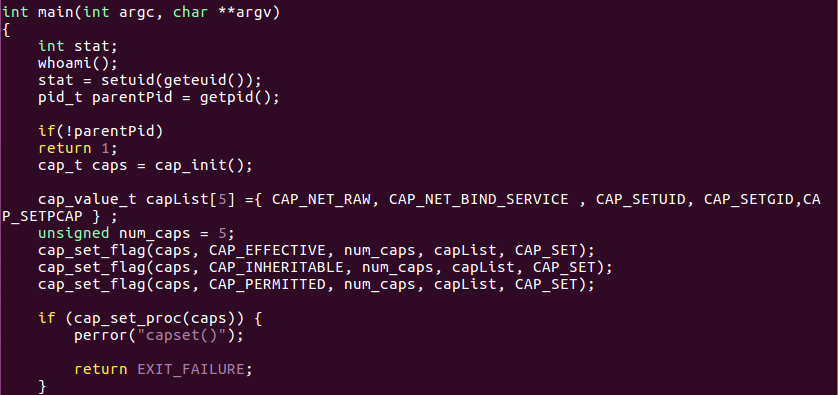
CAP\_LINUX\_IMMUTABLE : 9

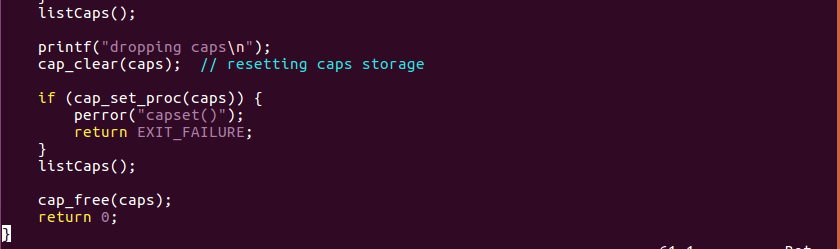
允许修改文件的不可修改(IMMUTABLE)权限和只添加(APPEND-ONLY)权限

该能力位关闭可以提高系统的安全性，但是同样系统的应用性能降低

1. 组合系统的部分能力位，实现系统的网络管理功能，或用户管理功能、文件管理功能。

代码如下：

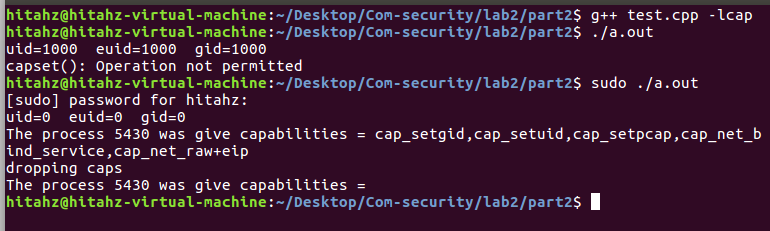




程序中设置了五种能力位：

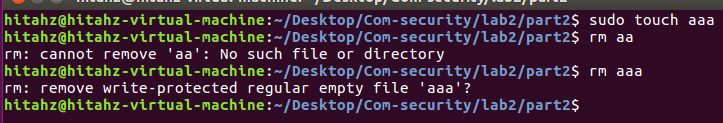


运行结果如下：

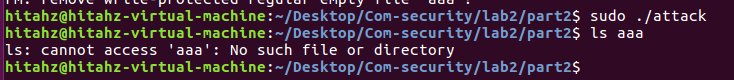


（4）编制攻击程序，测试能力位的安全性。

首先使用root权限创建文件aaa，普通用户无法删除



赋予当前文件夹cap\_chown能力位：运行程序删除aaa



代码如下：

