常的检测并插入一段代码造成的。如程序员可以在登录程序中插入一小段代码，让所有使用"zzzzz"登录名的用户成功登录而无论密码文件中的密码是什么。正常的程序代码如图9-22a所示。改成后门陷阱程序的代码如图9-22b所示。stremp这行代码的调用是为了判断登录名是否为"zzzzz"。如果是，则无论输入了什么密码都可以登录。如果后门陷阱被程序员放入到计算机生产商的产品中并飘洋过海，那么程序员日后就可以任意登录到这家公司生产的计算机上，而无论谁拥有它或密码是什么。后门陷阱程序的实质是它跳过了正常的认证过程。

```
while (TRUE) {
        printf("login: ");
        get_string(name);
        disable_echoing();
        printf("password: ");
        get_string(password);
        enable_echoing();
        v = check_validity(name, password);
        if (v) break;
}
execute_shell(name);
```

a)

```
while (TRUE) {
        printf("login: ");
        get_string(name);
        disable_echoing();
        printf("password: ");
        get_string(password);
        enable_echoing();
        v = check_validity(name, password);
        if (v || strcmp(name, "zzzzz") == 0) break;
}
execute_shell(name);
```

b)

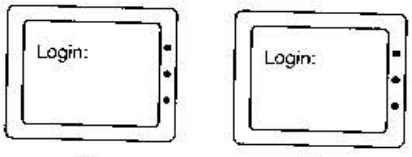
图9-22　a) 正常的代码；b) 插入了后门陷阱的代码

对公司来说，防止后门的一个方法是把代码审查（code review）作为标准惯例来执行。通过这一技术，一旦程序员完成对某个模块的编写和测试后，该模块被放入代码数据库中进行检验。开发小组里的所有程序员周期性地聚会，每个人在小组面前向大家解释每行代码的含义。这样做不仅增加了找出后门代码的机会，而且增加了大家的责任感，被抓出来的程序员也知道这样做会损害自己的职业生涯。如果该建议遭到了太多的反对，那么让两个程序员相互检查代码也是一个可行的方法。

9.5.3 登录欺骗

这种内部攻击的实施者是系统的合法用户，然而这些合法用户却试图通过登录欺骗的手段获取他人的密码。这种攻击通常发生在一个具有大量多用户公用计算机的局域网内。很多大学就有可以供学生使用的机房，学生可以在任意一台计算机上进行登录。登录欺骗（login spoofing）。它是这样工作的：通常当没有人登录到UNIX终端或局域网上的工作站时，会显示如图9-23a所示的屏幕。当用户坐下来输入登录名后，系统会要求输入口令。如果口令正确，用户就可以登录并启动shell（也有可能是GUI）程序。

现在我们来看一看这一情节。一个恶意的用户Mal写了一个程序可以显示如图9-23b所示的图像。除了内部没有运行登录程序外，它看上去和9-23a惊人的相似，这不过是骗人。现在Mal启动了他的程序，便可以躲在远处看好戏了。当用户坐下来输入登录名后，程序要求输入口令并屏蔽了响应。随后，登录名和口令后被写入文件并发出信号要求系统结束shell程序。这使得Mal能够正常退出登录并触发真正的登录程序，如图9-23a所示。好像是用户出现了一个拼写错误并要求再次登录，这时真正的登录程序开始工作了。但与此同时Mal又得到了另一对组合（登录名和口令）。通过在多个终端上进行登录欺骗，入侵者可收集到多个口令。



a)　b)

图9-23　a) 正确的登录屏幕；b) 假冒的登录屏幕

防止登录欺骗的惟一实用的办法是将登录序列与用户程序不能捕捉的键组合起来。Windows为此目的采用了Ctrl-Alt-Del。如果用户坐在终端前开始按Ctrl-Alt-Del，当前用户就会被注销并启动新的登录程序。没有任何办法可以跳过这一步。

9.6 利用代码漏洞

前面已经介绍了内部人员是如何危害系统安全的，在本节中，我们将介绍外部人员（outsider）（主要通过互联网）对操作系统进行攻击和破坏的方式。几乎所有的攻击机制都利用了操作系统或是被广泛使用的软件（如IE浏览器和微软Office）中的漏洞。一种典型的攻击形成方式是，有人发现了操作系统