则看不到红眼)。阿姆斯特丹机场从2001年起就开始使用虹膜识别技术以便使得经常出入机场的常客得以跳过常规安检流程。

还有一种技术叫做签名分析。用户使用一种特殊的笔签名，笔与终端相连。计算机将签名与在线存放的或智能卡里的已知样本进行比较。更好的一种办法是不去比较签名，而是比较笔的移动轨迹及书写签名时产生的压力。一个好的伪造者也许能够复制签名，但对笔画顺序和书写的压力与速度却毫无办法。

还有一种依靠迷你装置识别的技术是声音测定（Markowitz, 2000）。整个装置只需要一个麦克风（或者甚至是一部电话）和有关的软件即可。声音测定技术与声音识别技术不同。后者是为了识别人们说了些什么，而前者是为了判断人们的身份。有些系统仅仅要求用户说一句密码，但是窃听者可以把这句话录下来，通过回放来进入系统。更先进的系统向用户说一些话并要求重述，用户每次登录叙述的都是不同的语句。有些公司开始在软件中使用声音测定技术，如通过电话线连接使用的家庭购物软件。在这种情况下，声音测定比用PIN密码要安全得多。

我们可以继续给出许多例子，但是有两个例子特别有助于我们理解。猫和其他一些动物通过小便来划定自己的地盘。很明显，猫通过这种方法可以相互识别自己的家。假设某人拿着一个可以进行尿液分析的装置，那么他就可以建立识别样本。每个终端都可以有这样的装置，装置前放着一条标语："要登录系统，请留下样本。"这也许是一个绝对无法攻破的系统，但用户可能难以接受使用这样的系统。

在使用指纹识别装置和小型谱仪时也可能发生同样的情况。用户会被要求按下大拇指并抽取一滴血进行化验分析。问题在于任何验证识别系统对用户来说应该从心理上是可接受的。手指长度识别也许不会引起什么麻烦，但是类似于在线存储指纹等方式虽然减少了入侵的可能，但对大多数人来说是不可接受的。因为他们将指纹和犯人联系在一起。

9.5 内部攻击

前几节对于用户认证工作原理的一些细节问题已经有所讨论。不幸的是，阻止不速之客登录系统仅仅是众多安全问题中的一个。另一个完全不同的领域可以被定义为"内部攻击"（inside jobs），内部攻击由一些公司的编程人员或使用这些受保护的计算机、编制核心软件的员工实施。来自内部攻击与外部攻击的区别在于，内部攻击者拥有外部人员所不具备的专业知识和访问权限。下面我们将给出一些内部攻击的例子，这些攻击方式曾经非常频繁地出现在公司中。根据攻击者、被攻击者以及攻击者想要达到的目的这三方面的不同，每种攻击都具有不同的特点。

9.5.1 逻辑炸弹

在软件外包盛行的时代，程序员总是很担心他们会失去工作，有时候他们甚至会采取某些措施来减轻这种担心。对于感受到失业威胁的程序员，编写逻辑炸弹（logic bomb）就成为了一种策略。这一装置是某些公司程序员（当前被雇用的）写的程序代码，并被秘密地放入产品的操作系统中。只要程序员每天输入口令，产品就相安无事。但是一旦程序员被突然解雇并毫无警告地被要求离开时，第二天（或第二周）逻辑炸弹就会因得不到口令而发作。当然也可以在逻辑炸弹里设置多个变量。一个非常有名的例子是：逻辑炸弹每天核对薪水册。如果某程序员的工号没有在连续两个发薪日中出现，逻辑炸弹就发作了（Spafford等人, 1989）。

逻辑炸弹发作时可能会擦去磁盘，随机删除文件，对核心程序做难以发现的改动，或者对原始文件进行加密。在后面的例子中，公司对是否要叫警察带走放置逻辑炸弹的员工进退两难（报警存在着导致数月后对该员工宣判有罪的可能，但却无法恢复丢失的文件）。或者屈服该员工对公司的敲诈，将其重新雇用为"顾问"来避免如同天文数字般的补救，并依此作为解决问题的交换条件（公司也同时期望他不会再放置新的逻辑炸弹）。

在很多有记录的案例中，病毒向被其感染的计算机中植入逻辑炸弹。一般情况下，这些逻辑炸弹被设计为在未来的某个时间"爆炸"。然而，由于程序员无法预知那一台计算机将会被攻击，因此逻辑炸弹无法用于保护自己不失业，也无法用户勒索。这些逻辑炸弹通常会被设定为在政治上有重要意义的日子爆炸，因此它们也称做时间炸弹（time bomb）。

9.5.2 后门陷阱

另一个由内部人员造成的安全漏洞是后门陷阱（trap door）。这一问题是由系统程序员跳过一些通