对象被打开后，调用者会获得一个句柄。在后续的调用中，只需检查尝试的操作是否在打开时所申请的操作集合内，这样就避免了调用者为了读而打开文件然后对该文件进行写操作。另外，正如SACL所要求的那样，在句柄上进行的调用可能会导致产生审计日志。

Windows Vista增加了另外的安全设施来应对使用ACL保护系统的共同问题。进程的令牌中含有新增加的必需的完整性级别（Integrity-Level）SID字段并且对象在SACL中指定了一个完整性级别ACE。完整性级别阻止了对对象的写访问，不管DACL中有何种ACE。特别地，完整性级别方案用来保护系统免受被攻击者控制的Internet Explorer进程（可能用户接受了不妥的建议而从未知的网站下载代码）的破坏。低权限的IE，运行时的完整性级别被设置为低。系统中所有的文件和注册表中的键拥有中级的完整性级别，因此低完整性级别的IE不能修改它们。

近年来Windows增加了很多其他的安全特性。对于Windows XP Service Pack 2来说，系统的大部分在编译时使用了可对多种栈缓冲区溢出漏洞进行验证的选项（/GS）。另外，在AMD64体系结构中一种叫做NX的设施可限制执行栈上的代码。即使在x86模式下处理器中的NX位也是可用的。NX代表不可执行（no execute），它可以给页面加上标记使得其上的代码不能被执行。这样，即使攻击者利用缓冲区溢出漏洞向进程插入代码，跳转到代码处开始执行也不是一件容易的事情。

Windows Vista引入了更多的安全特性来阻止攻击者。加载到内核态的代码要经过检查（这在x64系统中是默认的）并且只有被正确签名的代码才可以被加载。在每个系统中，DLL和EXE的加载地址连同栈分配的地址都经过了有意的混排，这使得攻击者不太可能利用缓冲区溢出漏洞跳转到一个众所周知的地址然后执行一段被特意编排的可获得权限提升的代码。会有更小比例的系统受到依赖于标准地址处的二进制数据的攻击。在受到攻击时系统更加可能只是崩溃掉，将一个潜在的权限升级攻击转化为危险性更小的拒绝服务攻击。

在微软公司称为用户账户控制（User Account Control，UAC）的引入是另一个改变。这用来解决大部分用户以管理员身份运行系统这个长期的问题。Windows的设计并不需要用户以管理员身份使用系统，但在很多发布版本中对此问题的忽视使得如果你不是管理员就不可能顺利地使用Windows。始终以管理员身份使用系统是危险的。用户的错误会轻易地毁坏系统，而且如果用户由于某种原因被欺骗或攻击了而去运行可能危害系统的代码，这些代码将拥有管理员的访问权限并且可能会把其自身深深埋藏在系统中。

如果有UAC，当尝试执行需要管理员访问权限的操作时，系统会显示一个重叠的特殊桌面并且接管控制权，使得只有用户的输入可以授权这次访问（与C2安全中CTRL-ALT-DEL的工作方式类似）。当然，攻击者不需要成为管理员也可以破坏用户所真正关心的，比如他的个人文件。但UAC确实可阻止现有类型的攻击，并且如果攻击者不能修改任何系统数据或文件，那受损的系统恢复起来也比容易。

Windows Vista中最后的一个安全特性已经提到过了。这就是对具有安全边界的受保护进程（protected process）的支持。通常，在系统中用户（由令牌对象代表）定义了权限的边界。创建进程后，用户可通过任意数目的内核设施来访问进程以进行进程创建、调试、获取路径名和线程注入等。受保护进程关掉了用户的访问权限。这个设施在Vista中的唯一用处就是允许数字版权管理软件更好地保护内容。对受保护进程的使用在未来的发布版本中可能会用于对用户更加友好的目的，比方说保护系统以应对攻击者而不是保护内容免受系统所有者的攻击。

由于世界范围内越来越多的针对Windows系统的攻击，近年来微软公司加大了提高Windows安全性的努力。其中某些攻击非常成功，使得整个国家和主要公司的计算机都宕掉了，导致了数十亿美元的损失。这些攻击大都利用了编码中的小错误，这些错误可导致缓冲区溢出，从而使得攻击者可以通过重写返回地址、异常处理指针和其他数据来控制程序的执行。使用类型安全的语言而不是C和C++可避免许多此类的问题。即使使用这些不安全的语言，如果让学生更好地理解参数和数据验证中的陷阱，许多漏洞也可以避免。毕竟，许多在Microsoft编写代码的软件工程师在几年前也还是学生，就像正在阅读此实例研究的你们中的许多人一样。有许多关于在基于指针的语言中可被利用的编码上的小错误的类型以及怎样避免的书籍（比如，Howard和LeBlank，2007）。

11.10 小结

Windows Vista中的内核态由HAL、NTOS的内核和执行体层以及大量实现了从设备服务到文件系统、