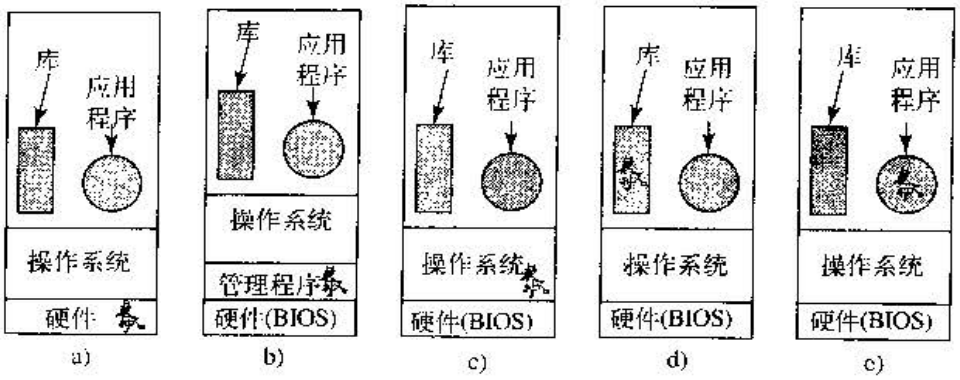

图9-30 rootkit可以隐藏的五种位置

2. rootkit检测

当硬件、操作系统、库和应用程序不能被信任时，rootkit很难被检测到。例如，一种查找rootkit的明显方法是列举磁盘上的所有文件，但是读取目录的系统调用、调用系统调用的库函数以及列表程序都有潜在的恶意性，并有可能忽略掉与rootkit相关的文件。然而情况也绝非无可救药。

检测一个引导自己的管理程序并在其控制下的虚拟机中运行操作系统和应用程序的rootkit虽然难以处理但也并非不可能。这要求从性能和功能上仔细检查虚拟机和实际机器的细微差异。Garfinkel等（2007）已经提出了一些这样的差异（如下所述），Carpenter等（2007）也讨论了这个话题。

一类检测方法依赖于一个事实：管理程序自身使用物理资源而失去这些资源可以被检测到。例如，管理程序需要使用一些TLB入口，在这些稀缺资源的使用上与虚拟机产生竞争。rootkit检测程序可以向TLB施加压力，观察其性能并与此前在裸机上测量的性能数据进行比较。

另一类检测方法与计时相关，尤其与虚拟输入输出设备的计时相关。假设在实际机器上读出一些PCI设备寄存器需要100个时钟周期，这个时间很容易重现。在一个虚拟环境下，这个寄存器的值来自于内存，它的读取时间依赖于它到底在CPU一级缓存、二级缓存还是实际RAM中。检测程序可以轻易地强迫其在这些状态之间来回移动并测量实际读取时间的变化。注意我们关注的是读取时间的变化而非实际的读取时间。

另一个可以被探查的部分是执行特权指令的时间，尤其是对那些在实际硬件上只需要几个时钟周期而在被模拟时需要几百或几千个时钟周期的特权指令。例如，如果读出某个被保护的CPU寄存器在实际硬件环境下需要1纳秒，那么10亿次软中断和模拟绝不可能在1秒内完成。当然，管理程序可以欺骗报告模拟时间而不报告所有涉及时间的系统调用的实际时间，检测程序可以通过连接提供精确时间基准的远程主机或网站来绕过时间模拟。因为检测程序只需要测量时间间隔（例如，执行10亿次被保护寄存器的读操作需要多少时间），本地时钟和远程时钟的偏移没有关系。

如果没有管理程序被塞入硬件和操作系统之间，那么rootkit可能被隐藏在操作系统中。很难通过引导计算机来检测其存在，因为操作系统是不可信的。例如，rootkit可能安装大量的文件，这些文件的文件名都由“$$$_”起始，当读取代表用户程序的目录时，不报告这些文件的存在。

在这样的环境下检测rootkit的一个方法是从一个可信的外部介质（如CD-ROM/DVD或USB棒）引导计算机，然后磁盘可以被一个反rootkit程序扫描，这时不用担心rootkit会干扰这个扫描。另一个选择是对操作系统中的每个文件做密码散列，这些散列值可以与一个列表中的散列值进行比较，这个列表在系统安装的时候生成并存储于系统外的一个不可被篡改的位置。如果没有预先建立这些散列值，也可以由安装CD-ROM或DVD即时计算得到，或由被比较文件自身进行计算得到。

库和应用程序中的rootkit更难隐藏，当操作系统从一个外部介质装入并可信时，这些库和应用程序的散列值也可以与已知为正确且存储与CD-ROM上的散列值进行比较。

到目前为止，我们讨论的都是被动rootkit，它们不会干扰检测软件。还存在一些主动rootkit，它们查找并破坏检测软件或至少将检测软件更改为永远报告“NO ROOTKITS FOUND!”（没有发现rootkit），这些rootkit要求更复杂的检测方法。幸运的是，到目前为止在现实环境下主动rootkit还没有出现。

在发现rootkit后应该做什么这个问题上存在两种观点。一种观点认为系统管理员应该像处理癌症的外科医生那样非常小心地切除它。另一种观点认为尝试移除rootkit太过危险，可能还有其他碎片隐藏在