Linux系统环境下的高级隐藏技术介绍 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/253/2021_2022_Linux_E7_B3_BB_E7_BB_c103_253041.htm 隐藏技术在计算机系统安全中应用十分广泛，尤其是在网络攻击中，当攻击者成功侵入一个系统后，有效隐藏攻击者的文件、进程及其加载的模块变得尤为重要。本文将讨论Linux系统中文件、进程及模块的高级隐藏技术，这些技术有的已经被广泛应用到各种后门或安全检测程序之中，而有一些则刚刚起步，仍然处在讨论阶段，应用很少。 1.隐藏技术 1.1.Linux下的中断控制及系统调用 Intel x86系列微机支持256种中断，为了使处理器比较容易地识别每种中断源，把它们从0~256编号，即赋予一个中断类型码n,Intel把它称作中断向量。 Linux用一个中断向量(128或者0x80)来实现系统调用，所有的系统调用都通过唯一的入口system_call来进入内核，当用户动态进程执行一条int 0x80汇编指令时，CPU就切换到内核态，并开始执行system_call函数，system_call函数再通过系统调用表sys_call_table来取得相应系统调用的地址进行执行。系统调用表sys_call_table中存放所有系统调用函数的地址，每个地址可以用系统调用号来进行索引，例如sys_call_table[NR_fork]索引到的就是系统调用sys_fork()的地址。 Linux用中断描述符(8字节)来表示每个中断的相关信息，其格式如下： 偏移量31…16 一些标志、类型码及保留位 段选择符 偏移量15…0 所有的中断描述符存放在一片连续的地址空间中，这个连续的地址空间称作中断描述符表(IDT)，其起始地址存放在中断描述符表寄存器(IDTR)中，其格式如下： 32位基址值 界限 其中各个结构的相应联系

可以如下表示： 通过上面的说明可以得出通过IDTR寄存器来找到system_call函数地址的方法：根据IDTR寄存器找到中断描述符表，中断描述符表的第0x80项即是system_call函数的地址，这个地址将在后面的讨论中应用到。 1.2.Linux 的LKM(可装载内核模块)技术 为了使内核保持较小的体积并能够方便的进行功能扩展，Linux系统提供了模块机制。模块是内核的一部分，但并没有被编译进内核，它们被编译成目标文件，在运行过程中根据需要动态的插入内核或者从内核中移除。由于模块在插入后是作为Linux内核的一部分来运行的，所以模块编程实际上就是内核编程，因此可以在模块中使用一些由内核导出的资源，例如Linux2.4.18版以前的内核导出系统调用表(sys_call_table)的地址，这样就可以根据该地址直接修改系统调用的入口，从而改变系统调用。在模块编程中必须存在初始化函数及清除函数，一般情况下，这两个函数默认为init_module()以及clearup_module()，从2.3.13内核版本开始，用户也可以给这两个函数重新命名，初始化函数在模块被插入系统时调用，在其中可以进行一些函数及符号的注册工作，清除函数则在模块移除系统时进行调用，一些恢复工作通常在该函数中完成。 1.3.Linux下的内存映像 /dev/kmem是一个字符设备，是计算机主存的映像，通过它可以测试甚至修改系统，当内核不导出sys_call_table地址或者不允许插入模块时可以通过该映像修改系统调用，从而实现隐藏文件、进程或者模块的目的。 1.4.proc 文件系统 proc文件系统是一个虚拟的文件系统，它通过文件系统的接口实现，用于输出系统运行状态。它以文件系统的形式，为操作系统本身和应用进程之间的通信提供了一个界面，使应用程序能够安全、方

便地获得系统当前的运行状况何内核的内部数据信息，并可以修改某些系统的配置信息。由于proc以文件系统的接口实现，因此可以象访问普通文件一样访问它，但它只存在于内存之中。 2.技术分析 2.1 隐藏文件 Linux系统中用来查询文件信息的系统调用是sys_getdents，这一点可以通过strace来观察到，例如strace ls 将列出命令ls用到的系统调用，从中可以发现ls是通过sys_getedents来执行操作的。当查询文件或者目录的相关信息时，Linux系统用sys_getedents来执行相应的查询操作，并把得到的信息传递给用户空间运行的程序，所以如果修改该系统调用，去掉结果中与某些特定文件的相关信息，那么所有利用该系统调用的程序将看不见该文件，从而达到了隐藏的目的。首先介绍一下原来的系统调用，其原型为：int sys_getdents(unsigned int fd, struct dirent *dirp,unsigned int count) 其中fd为指向目录文件的文件描述符，该函数根据fd所指向的目录文件读取相应dirent结构，并放入dirp中，其中count为dirp中返回的数据量，正确时该函数返回值为填充到dirp的字节数。下图是修改后的系统调用hacked_getdents执行流程。 图 系统调用hacked_getdents执行流程 