linux认证:分享Linux操作系统下隐藏文件Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/583/2021_2022_linux_E8_AE_A4_E8_AF_c103_583319.htm 一. 概述 目前通用的隐藏文件方法还是hooksys_getdents64系统调用， 大致流程就是先调用原始的sys_getdents64系统调用，然后在在buf中做过滤。修改sys_call_table是比较原始的rk技术了，碰到好点的管理员，基本上gdb一下vmlinux就能检测出来。 如何想做到更加隐蔽的话，就要寻找新的技术。 inline hook也是目前比较流行的做法，不容易检测。本文通过讲解一种利用inline hook内核中某函数， 来达到隐藏文件的方法。 二. 剖析sys_getdnts64系统调用 想隐藏文件， 还是要从sys_dents64系统调用下手。 去看下它在内核中是如何实现的。 代码在linux-2.6.26/fs/readdir.c中：asmlinkage long sys_getdents64(unsigned int fd, struct linux_dirent64 __user * dirent, unsigned int count) { struct file * file. struct linux_dirent64 __user * lastdirent. struct getdents_callback64 buf. int error. error = -EFAULT. if (!access_ok(VERIFY_WRITE, dirent, count)) goto out. error = -EBADF. file = fget(fd). if (!file) goto out. buf.current_dir = dirent. buf.previous = NULL. buf.count = count. buf.error = 0. error = vfs_readdir(file, filldir64, lt. 0) goto out_putf. error = buf.error. lastdirent = buf.previous. if (lastdirent) { typeof(lastdirent-gt.f_pos. error = -EFAULT. if (__put_user(d_off, gt.d_off)) goto out_putf. error = count - buf.count. } out_putf: fput(file). out: return error. } 首先调用access_ok来验证是下用户空间的dirent地址是否越界，是否可写。 接着根据fd，利用fget找到对应的file结构。 接着出现了一个填充buf数据结构

的操作，先不管它是干什么的，接着往下看。 vfs_readdir(file, filldir64, gt.f_path.dentry-gt.f_op || !file-gt.readdir) goto out. res = security_file_permission(file, MAY_READ). if (res) goto out. res = mutex_lock_killable(gt.i_mutex). if (res) goto out. res = -ENOENT. if (!IS_DEADDIR(inode)) { res = file-gt.readdir(file, buf, filler). file_accessed(file). } mutex_unlock(gt.i_mutex). out: return res. } EXPORT_SYMBOL(vfs_readdir).