linux驱动读取其他进程内存,开源一个Linux进程内存内核管理模块源码

转载

葛店小学张洪雨 于 2021-04-30 16:28:00 发布 265 收藏
文章标签： linux驱动读取其他进程内存
看完以上的逻辑。大家是不是柳暗花明又一村，心里开朗了许多，他们之间是可以相互转换的。

通过进程pid_t可以拿到pid，通过pid可以拿到task_struct。又可以反过来通过task_struct拿到进程pid。

关闭进程接口

驱动源码是使用put_pid将进程pid*的使用次数减去1。

在Linux内核源码/kernel/pid.c下可以看到：

void put_pid(struct pid *pid){struct pid_namespace *ns;if(!pid)return;ns = pid->numbers[pid->level].ns;if((atomic_read(&pid->count) == 1) ||atomic_dec_and_test(&pid->count)) {kmem_cache_free(ns->pid_cachep, pid);put_pid_ns(ns);}}EXPORT_SYMBOL_GPL(put_pid);

读、写进程内存接口

首先根据pid*用get_pid_task取出task_struct。再用get_task_mm取出mm_struct结构。因为这个结构包含了进程的内存信息。首先检查内存是否可读if (vma->vm_flags & VM_READ)。

如果可读。那么开始计算物理内存地址位置。由于Linux内核默认开启MMU机制，所以只能以页为单位计算物理内存地址。计算物理内存地址的方法有很多。

如pagemap、pgd pud pmd pte、get_user_pages驱动里演示pagemap。

其他方法可自行参考Linux内核源码/fs/proc/task_mmu.c。

知道了物理内存地址后，读、写物理内存地址，Linux内核也有演示： drivers/char/mem.c。写的非常详细。最后还要注意MMU机制的离散内存，即buffer不连续问题。

获取进程内存块列表

这个接口没什么技术含量，都是照抄Linux内核源码的代码，fsproctask_mmu.c。

核心思想是通过task_struct取出mm_struct，接下来在mm_struct中遍历取出vma。

structmm_struct{structvm_area_struct* mmap;/* list of VMAs */structrb_rootmm_rb;structvm_area_struct* mmap_cache;/* last find_vma result */# ifdefCONFIG_MMUunsignedlong(*get_unmapped_area)(struct file *filp,unsignedlongaddr, unsignedlonglen,unsignedlongpgoff, unsignedlongflags);void(*unmap_area) (struct mm_struct *mm, unsignedlongaddr);# endifunsignedlongmmap_base; /* base of mmap area */unsignedlongmmap_legacy_base; /* base of mmap area in bottom-up allocations */unsignedlongtask_size; /* size of task vm space */unsignedlongcached_hole_size; /* if non-zero, the largest hole below free_area_cache */unsignedlongfree_area_cache; /* first hole of size cached_hole_size or larger */unsignedlonghighest_vm_end; /* highest vma end address */pgd_t* pgd;atomic_tmm_users; /* How many users with user space? */atomic_tmm_count; /* How many references to "struct mm_struct" (users count as 1) */intmap_count; /* number of VMAs */spinlock_tpage_table_lock; /* Protects page tables and some counters */structrw_semaphoremmap_sem;structlist_headmmlist;/* List of maybe swapped mm's. These are globally strung* together off init_mm.mmlist, and are protected* by mmlist_lock*/unsignedlonghiwater_rss; /* High-watermark of RSS usage */unsignedlonghiwater_vm; /* High-water virtual memory usage */unsignedlongtotal_vm; /* Total pages mapped */unsignedlonglocked_vm; /* Pages that have PG_mlocked set */unsignedlongpinned_vm; /* Refcount permanently increased */unsignedlongshared_vm; /* Shared pages (files) */unsignedlongexec_vm; /* VM_EXEC & ~VM_WRITE */unsignedlongstack_vm; /* VM_GROWSUP/DOWN */unsignedlongdef_flags;unsignedlongnr_ptes; /* Page table pages */unsignedlongstart_code, end_code, start_data, end_data;unsignedlongstart_brk, brk, start_stack;unsignedlongarg_start, arg_end, env_start, env_end;unsignedlongsaved_auxv[AT_VECTOR_SIZE]; /* for /proc/PID/auxv */

获取进程命令行

mm_struct结构体里面有个arg_start变量，储存的地址值即进程命令行

为了避免不法分子将此驱动用在非法的用途， 在此给出侦测建议：

1. 检查是否有/dev/rwProcMem33此文件。

2. 检查SELinux是否被关闭。从安卓5.0启用SELinux后，APP想要与驱动进行通讯，必须得关闭SELinux，如果发现SELinux是关闭状态，并且是高版本的安卓系统，那么此安卓使用者必有问题。

检测SELinux的方法很多，如open打开某个文件、ioctl等等，如果SELinux是打开状态，那么这些都会直接返回EACCES (Permission denied)，提示拒绝访问。

另外，在高版本的安卓系统中，如安卓10，如果SELinux是打开的状态，那么lsmod查看驱动列表，也会直接返回Permission denied拒绝访问，如果lsmod能直接显示驱动列表，那么此使用者的SELinux也是有问题的，有可能被关闭了的。

最后开源地址(含demo)

Github链接： https://github.com/abcz316/rwProcMem33

总结

首先，编译此源码需要一定的技巧，再者，手机在出厂时本身已设置多重障碍用来阻止第三方驱动的加载(即使你拥有root权限也无法加载)，此源码仅供交流学习Linux系统使用。

看雪ID：abcz316

*这里由看雪论坛 abcz316 原创，转载请注明来自看雪社区。

好书推荐

﹀

﹀

﹀

戳“阅读原文”一起来充电吧！返回搜狐，查看更多