**LKM（Loadable Kernel Module）**

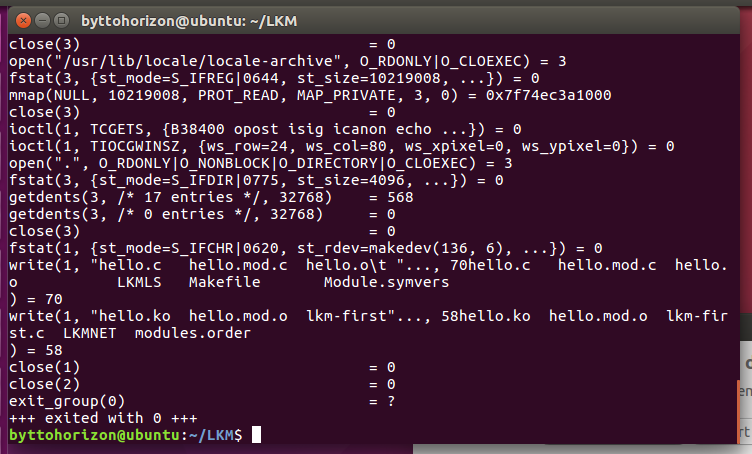
**Linux可加载内核模块，通过运行在内核层面的代码在系统运行的过程中动态更改Linux。**

**在隐藏层面，管理员可能会调用的命令有ls、ps、netstat等，与之相关的重要系统调用有sys\_getgends、sys\_query\_module、sys\_getuid等；在攻击层面，文件的读写则涉及到sys\_write、sys\_read、sys\_open等一系列系统调用。**

**LKM编程通过替换或者修改上述系统调用，使系统命令对特定恶意程序无视，或者对用户读写进行记录，可以隐蔽地达成期望的攻击效果。**

这一阶段编程实现的内核模块包括

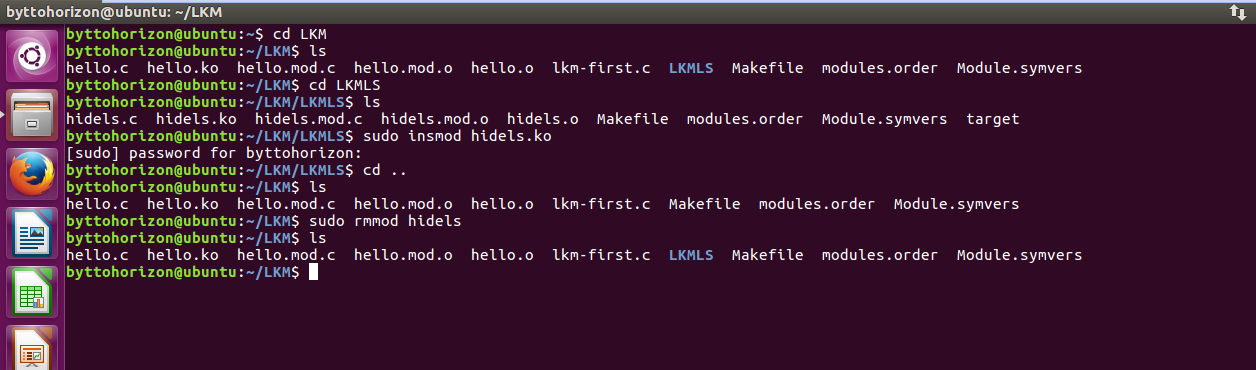
1. ls命令下基于文件名的隐藏



输入命令strace，可以查看命令相关的系统调用，上图为strace ls的结果

可以看到ls命令的核心系统调用为getdents。

通过对该调用进行劫持，过滤相关文件信息后，把过滤后的信息重新写回，即可使系统ls命令的执行过程中实现相关文件信息的过滤。实际效果如下



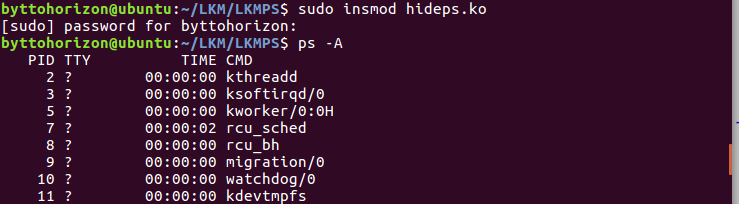
可以看到在加载模块hidels后，LKM目录下的LKMLS文件信息被成功隐藏，卸载模块后文件成功出现。

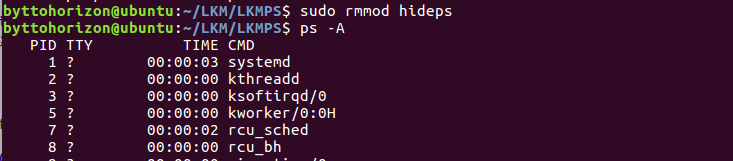
在目录被隐藏时，相关文件依然存在，仍然可以通过cd等方法进入目录进行操作。

1. ps命令下基于进程名的隐藏

ps命令与ls类似，只不过可以匹配的信息从文件名变为了进程名或是PID。

实际效果如下





Linux系统下pid = 1的进程固定为systemd，作为超级进程负责产生其他所有进程，用于把整个系统带入可操作状态，在完成过滤后，可以看到pid = 1的条目在ps –A命令下被隐藏。

1. netstat命令下基于端口的隐藏

向用户态隐藏端口，实质是在用户进程读/proc下面的相关文件获取端口信息时，把需要隐藏的的端口的内容过滤掉，使得用户进程读到的内容里面没有想隐藏的端口。

再具体的操作过程中，我的选择是钩 /proc/net/tcp 等文件的 show 函数，该函数用于展示序列中的信息，通过对这一函数的更改。根据端口信息，过滤一条/proc/net/tcp下的特定记录。

实际效果如下



根据系统情况选择隐藏53端口，最终结果是在netstat命令下隐藏tcp连接对于53号端口的侦听信息。

1. 内核模块自身的隐藏

lsmod命令通过/proc/modules来获取当前系统模块信息。而/proc/modules中的当前系统模块信息是内核利用struct modules结构体的表头遍历内核模块链表、从所有模块的struct module结构体中获取模块的相关信息来得到的。

在模块载入时，最简单的方法，通过调用list\_del\_init（struct list\_head \* entry）即可在链表中删去某一节点。

效果如下



载入模块成功后lsmod命令不会再发现该模块

但是该方法存在一个较明显的缺点，即在链表中删去模块信息后无法使用rmmod命令移除该内核模块，编程调试极为不便。