这是一道内核题目

常规操作当然是先ssh

```
[    2.737289] Write protecting the kernel read-only data: 2456k
[    2.922201] rootkit: module license 'unspecified' taints kernel.
[    2.922571] Disabling lock debugging due to kernel taint
/ # ls
bin         etc         lib         lost+found  rootkit.ko  tmp         var
dev         flag        linuxrc     proc        sbin        usr
/ # cat flag
[    5.890248] You will not see the flag...
cat: can't open 'flag': Operation not permitted
/ #
```

直接尝试cat flag 发现无法拿到flag

下载文件用ida查看

```
  sys_open = (int (__cdecl *)(_DWORD, _DWORD, _DWORD))MEMORY[0xC15FA034];
  sys_openat = (int (__cdecl *)(_DWORD, _DWORD, _DWORD, _DWORD))MEMORY[0xC15FA4BC];
  sys_symlink = (int (__cdecl *)(_DWORD, _DWORD))MEMORY[0xC15FA16C];
  sys_symlinkat = (int (__cdecl *)(_DWORD, _DWORD, _DWORD))MEMORY[0xC15FA4E0];
  sys_link = (int (__cdecl *)(_DWORD, _DWORD))MEMORY[0xC15FA044];
  sys_linkat = (int (__cdecl *)(_DWORD, _DWORD, _DWORD, _DWORD))MEMORY[0xC15FA4DC];
  sys_rename = (int (__cdecl *)(_DWORD, _DWORD))MEMORY[0xC15FA0B8];
  sys_renameat = (int (__cdecl *)(_DWORD, _DWORD, _DWORD))MEMORY[0xC15FA4D8];
  wp(0);
  v0 = (_DWORD *)sct;
  *(_DWORD *)(sct + 20) = sys_open_hooked;
  v0[295] = sys_openat_hooked;
  v0[83] = sys_symlink_hooked;
  v0[304] = sys_symlinkat_hooked;
  v0[9] = sys_link_hooked;
  v0[303] = sys_linkat_hooked;
  v0[38] = sys_rename_hooked;
  v0[302] = sys_renameat_hooked;
  wp(1);
```

发现这是ko文件（linux内核模块文件）

主要操作是hook了sys_open等一类可以打开文件的系统调用。

```
ext:08000257
ext:08000257 loc_8000257:                              ; DATA XREF: __mcount_loc:0800040C↓o
ext:08000257                    call    mcount
ext:0800025C                    mov     edx, offset aFlag ; "flag"
ext:08000261                    mov     ebx, [ebp+arg_0]
ext:08000264                    mov     eax, ebx
ext:08000266                    call    strstr
ext:0800026B                    test    eax, eax
ext:0800026D                    jnz     short loc_800028C
ext:0800026F                    mov     eax, [ebp+arg_8]
ext:08000272                    mov     [esp], ebx      ; _DWORD
ext:08000275                    mov     [esp+8], eax    ; _DWORD
ext:08000279                    mov     eax, [ebp+arg_4]
ext:0800027C                    mov     [esp+4], eax    ; _DWORD
ext:08000280                    call    ds:sys_open
ext:08000286
ext:08000286 loc_8000286:                              ; CODE XREF: sys_open_hooked+4B↓j
ext:08000286                    add     esp, 0Ch
```

每个函数大概都是这种形式的，如果操作的文件名是包含flag，就啥也不让你干，其他随意（这意味着我们也可以修改rootkit.ko文件）

解决办法

```
sed -i rootkit.ko -e 's/rootkit/wsxkhhh/g'
sed -i rootkit.ko -e 's/flag/wsxk/g'
sed 's/\xa1\x34\xa0\x5f\xc1/\xb8\x70\x8d\x15\xc1/g' -i rootkit.ko
```

这些操作都在本地进行（下载的rootkit重命名位rootkit.ko即可）

```
base64 rootkit.ko
```

使用base64命令后把屏幕的内容复制然后粘贴到服务机上（命名为1.txt）

在服务机上使用以下命令

```
cat 1.txt | base64 -d > 2.ko
insmod 2.ko
```

接下来即可cat flag
注意：flag文件是个压缩包（gzip文件）
同样用base64的方法拷到本地上解压就能拿到flag

感谢：
https://lyq.blogd.club/2021/11/07/rootkit-writeup/
https://aufarg.github.io/pwnablekr-rootkit-400.html