```
 ____  __   __  ____   ____  ____  _     ___     __  _  ____
|    \|  |__|  ||    \ /    ||    \| |   /  _]   |  |/ ]|    \
|  o  )  |  |  ||  _  ||  o  ||  o  )| |  /  [_    |  ' / |  D  )
|   _/|  |  |  ||  |  ||     ||     || |___|    _]   |    \ |    /
|  | |  `  '  ||  |  ||  _  ||  O  ||     ||   [_ __ |     \|    \
|  |  \      / |  |  ||  |  ||     ||     ||  ||  .  ||  .  \
|__|   \_/\_/  |__|__||__|__||_____||_____||_____||__||__|\_||__|\_|
```

- Site admin : daehee87.kr@gmail.com
- IRC : irc.smashthestack.org:6667 / #pwnable.kr
- Simply type "irssi" command to join IRC now
- files under /tmp can be erased anytime. make your directory under /tmp

Last login: Thu May 5 11:16:29 2016 from 188.86.45.253
random@ubuntu:~$ ls
flag random random.c

처음 ssh**로 접속을 하면** 3**개의 파일이있다**.

random@ubuntu:~$ ls -la
total 36
drwxr-x--- 4 root   random 4096 Aug 20 2014 .
dr-xr-xr-x 61 root   root   4096 Mar 7 02:28 ..
d--------- 2 root   root   4096 Jun 30 2014 .bash_history
-r--r----- 1 random2 root   49 Jun 30 2014 flag
dr-xr-xr-x 2 root   root   4096 Aug 20 2014 .irssi
-r-sr-x--- 1 random2 random 8538 Jun 30 2014 random
-rw-r--r-- 1 root   root   301 Jun 30 2014 random.c

파일을 자세하게 보면 우리가 열어야 하는 파일은 flag**라는 파일이다**. **하지만** flag**라는 파일은 파일의 제작자인** random2 **또는** root**그룹에 속해 있어야 한다**. **하지만 두곳에도 해당 되지 않으므로** random**을 실행시켜** random2**의 권한을 얻으려고 한다**. **관리자가 친절하게** Random.c **소스 코드를 만들어 뒀지만**. **나는 어셈블리어를 배우고 있기에** c**언어로 바꾸며 풀어보았다**.

Dump of assembler code for function main:
   0x00000000004005f4 <+0>:   push   rbp
   0x00000000004005f5 <+1>:   mov   rbp,rsp
   0x00000000004005f8 <+4>:   sub   rsp,0x10
   0x00000000004005fc <+8>:   mov   eax,0x0
   0x0000000000400601 <+13>:   call   0x400500 <rand@plt>
   0x0000000000400606 <+18>:   mov   DWORD PTR [rbp-0x4],eax
   0x0000000000400609 <+21>:   mov   DWORD PTR [rbp-0x8],0x0
   0x0000000000400610 <+28>:   mov   eax,0x400760
   0x0000000000400615 <+33>:   lea   rdx,[rbp-0x8]
   0x0000000000400619 <+37>:   mov   rsi,rdx
   0x000000000040061c <+40>:   mov   rdi,rax
   0x000000000040061f <+43>:   mov   eax,0x0
   0x0000000000400624 <+48>:   call   0x4004f0 <__isoc99_scanf@plt>
   0x0000000000400629 <+53>:   mov   eax,DWORD PTR [rbp-0x8]
   0x000000000040062c <+56>:   xor   eax,DWORD PTR [rbp-0x4]
   0x000000000040062f <+59>:   cmp   eax,0xdeadbeef
   0x0000000000400634 <+64>:   jne   0x400656 <main+98>

```
   0x0000000000400636 <+66>:    mov    edi,0x400763
   0x000000000040063b <+71>:    call   0x4004c0 <puts@plt>
   0x0000000000400640 <+76>:    mov    edi,0x400769
   0x0000000000400645 <+81>:    mov    eax,0x0
   0x000000000040064a <+86>:    call   0x4004d0 <system@plt>
   0x000000000040064f <+91>:    mov    eax,0x0
   0x0000000000400654 <+96>:    jmp    0x400665 <main+113>
   0x0000000000400656 <+98>:    mov    edi,0x400778
   0x000000000040065b <+103>:   call   0x4004c0 <puts@plt>
   0x0000000000400660 <+108>:   mov    eax,0x0
   0x0000000000400665 <+113>:   leave
   0x0000000000400666 <+114>:   ret
```

위의 코드를 c로 바꾸어봤다.

```
#include <stdio.h>

int main()
{
	int v4, v8;

	v4 = rand();
	scanf("%d", &v8);
	if((v8 ^ v4) == 0xdeadbeef)
	{
		puts("Good!");
		system("/bin/cat flag");
	}
	else
	{
		puts("Wrong, maybe you should try 2^32 cases.");
	}
	return 0;
}
```

rand의 값은 seed값을 줘야 발생하는 수가 달라지는데 seed값을 안줘서 그런지 계속 같다.. 뭐징..
어차피 우리는 v8의 값을찾아서 0xdeadbeef와 비트가 같기만 하면 되기때문에 (v8 ^ v4)에 ^ v4를 추가하고 0xdeadbeef에 ^ v4를 추가시켜 v8의 값을 찾을수 있었다. 덧셈 뺄셈하듯 같은 수의 xor을 양 변에 추가해도 값이 똑같다고 한다.

xor을 시킨 후 값을 입력해보면 Good!이 뜬다 오예!