```
 ____  __   __  ____   ____  ____  _      ___      __  _  ____
|    \|  |__|  ||    \ /    ||    \| |     /  _]    |  |/ ]|    \
|  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 08:42:04 2016 from 220.80.17.96
fd@ubuntu:~$ ls
fd fd.c flag

처음에 접속하면 저 3개의 파일이 있는데,

fd@ubuntu:~$ ls -al
total 32
drwxr-x--- 4 root fd   4096 Aug 20 2014 .
dr-xr-xr-x 61 root root 4096 Mar 7 02:28 ..
d--------- 2 root root 4096 Jun 12 2014 .bash_history
-r-sr-x--- 1 fd2 fd   7322 Jun 11 2014 fd
-rw-r--r-- 1 root root 418 Jun 11 2014 fd.c
-r--r----- 1 fd2 root   50 Jun 11 2014 flag
dr-xr-xr-x 2 root root 4096 Aug 20 2014 .irssi
fd@ubuntu:~$ whoami
fd
fd@ubuntu:~$ id
uid=1001(fd) gid=1001(fd) groups=1001(fd)

flag**파일을 열기위해선** fd**파일을 열어** fd2**의 권한이 필요했다.**
하지만 나는 지금 어셈블리 공부를 하고 있기 때문에 gdb**로 파일을 열어 어셈블리어를 보며** c**로 바꾸어 보았다.**

(gdb) disas main
Dump of assembler code for function main:
   0x08048494 <+0>: push   ebp
   0x08048495 <+1>: mov   ebp,esp
   0x08048497 <+3>: push   edi
   0x08048498 <+4>: push   esi
   0x08048499 <+5>: and   esp,0xfffffff0
   0x0804849c <+8>: sub   esp,0x20
   0x0804849f <+11>: cmp   DWORD PTR [ebp+0x8],0x1
   0x080484a3 <+15>:   jg   0x80484bb <main+39>
   0x080484a5 <+17>:   mov   DWORD PTR [esp],0x8048630
   0x080484ac <+24>:call   0x8048380 <puts@plt>
   0x080484b1 <+29>:   mov   eax,0x0
   0x080484b6 <+34>:   jmp   0x8048559 <main+197>
   0x080484bb <+39>:   mov   eax,DWORD PTR [ebp+0xc]
   0x080484be <+42>:   add   eax,0x4

```
  0x080484c1 <+45>:    mov    eax,DWORD PTR [eax]
  0x080484c3 <+47>:    mov    DWORD PTR [esp],eax
  0x080484c6 <+50>:    call   0x80483d0 <atoi@plt>
  0x080484cb <+55>:    sub    eax,0x1234
  0x080484d0 <+60>:    mov    DWORD PTR [esp+0x18],eax
  0x080484d4 <+64>:    mov    DWORD PTR [esp+0x1c],0x0
  0x080484dc <+72>:    mov    DWORD PTR [esp+0x8],0x20
  0x080484e4 <+80>:    mov    DWORD PTR [esp+0x4],0x804a060
  0x080484ec <+88>:mov    eax,DWORD PTR [esp+0x18]
  0x080484f0 <+92>:mov    DWORD PTR [esp],eax
  0x080484f3 <+95>:call   0x8048370 <read@plt>
  0x080484f8 <+100>:   mov    DWORD PTR [esp+0x1c],eax
  0x080484fc <+104>:   mov    edx,0x8048646
  0x08048501 <+109>:   mov    eax,0x804a060
  0x08048506 <+114>:   mov    ecx,0xa
  0x0804850b <+119>:   mov    esi,edx
  0x0804850d <+121>:   mov    edi,eax
  0x0804850f <+123>:   repz cmps BYTE PTR ds:[esi],BYTE PTR es:[edi]
  0x08048511 <+125>:   seta   dl
  0x08048514 <+128>:   setb   al
  0x08048517 <+131>:   mov    ecx,edx
  0x08048519 <+133>:   sub    cl,al
  0x0804851b <+135>:   mov    eax,ecx
  0x0804851d <+137>:   movsx  eax,al
  0x08048520 <+140>:   test   eax,eax
  0x08048522 <+142>:   jne    0x8048548 <main+180>
  0x08048524 <+144>:   mov    DWORD PTR [esp],0x8048650
  0x0804852b <+151>:   call   0x8048380 <puts@plt>
  0x08048530 <+156>:   mov    DWORD PTR [esp],0x804865c
  0x08048537 <+163>:   call   0x8048390 <system@plt>
  0x0804853c <+168>:   mov    DWORD PTR [esp],0x0
  0x08048543 <+175>:   call   0x80483b0 <exit@plt>
  0x08048548 <+180>:   mov    DWORD PTR [esp],0x804866a
  0x0804854f <+187>:   call   0x8048380 <puts@plt>
  0x08048554 <+192>:   mov    eax,0x0
  0x08048559 <+197>:   lea    esp,[ebp-0x8]
  0x0804855c <+200>:   pop    esi
---Type <return> to continue, or q <return> to quit---
  0x0804855d <+201>:   pop    edi
  0x0804855e <+202>:   pop    ebp
  0x0804855f <+203>:   ret
End of assembler dump.
```

위의 어셈블리 코드를 c언어로 바꾸어 보았다.

```
char buf[32] = {0};

int main(int argc, char *argv[])
{
        int v18, v1c;
        if(argc > 1)
        {
```

```
		v18 = atoi(argv[1]) - 0x1234;
		v1c = read(v18, buf, 0x20);
		if(strcmp("LETMEWIN", buf) == 0)
		{
			puts("good job :)");
			system("/bin/cat flag");

		}
		else
		{
			puts("learn about Linux file IO");
		}
	}
	else
	{
		puts("pass argv[1] a number");
	}
	return 0;
}
```

우리의 최종 목표는 LETMEWIN과 buf의 값이 둘다 같아야 하므로 buf값을 조작 해야 하므로 read 함수에 들어 있는 v18의 값을 0 으로 만들어 입력을 할 수 있게 만들어야 한다. argv[1]의 값이 0x1234여야 0이 되므로 argv[1]은 4660이 되어 v18을 0으로 만들어 입력을 할 수 있게 만든다. 이로써 값을 입력해보면

```
fd@ubuntu:~$ ./fd 4660
LETMEWIN

good job :)
```

fd파일을 열어 flag파일을 열 수 있었다.