	fd - 1 pt [writeup]
ſ	Mommy! what is a file descriptor in Linux?
t	try to play the wargame your self but if you are ABSOLUTE beginner, follow this utorial link: https://youtu.be/971eZhMHQQw
s	ssh fd@pwnable.kr -p2222 (pw:guest)
	mumand (22005) times a comb 20 mumans and 1 mu for it
	pwned (32995) times. early 30 pwners are: go_for_it v
	Flag?: auth

우분투 terminal에 ssh fd@pwnable.kr -p2222로 들어간다.(비밀번호:guest)



ls -al로 들어가서 각 파일별 소유자, 소유그룹, 파일권한을 살핀다. fd 계정으로 접속했으니 fd 파일은 소유 그룹이 fd이므로 읽기, 실행이 가능하다. fd.c 파일은 소유자, 소유그룹이 둘다 root이므로 other의 권한인 읽기만 가능하다. flag 파일은 마찬가지로 other에 속하기 때문에 other의 권한으로 접근이 가능한데, 권한이 없다.

```
fd@pwnable:~$ ls -l
total 16
-r-sr-x--- 1 fd_pwn fd 7322 Jun 11 2014 fd
-rw-r--r-- 1 root root 418 Jun 11 2014 fd.c
-r--r----- 1 fd_pwn root 50 Jun 11 2014 flag
```

fd.c를 가지고 fd elf 파일을 만들었을 확률이 높으므로 f.dc를 분석하기위해 vim fd.c를 입력하다.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char buf[32];
int main(int argc, char* argv[], char* envp[]){
         if(argc<2){
                  printf("pass argv[1] a number\n");
                  return 0:
         int fd = atoi( argv[1] ) - 0x1234;
         int len = 0;
         len = read(fd, buf, 32);
         if(!strcmp("LETMEWIN\n", buf)){
    printf("good job :)\n");
    system("/bin/cat flag");
                  exit(0);
         printf("learn about Linux file IO\n");
         return 0;
}
"fd.c" [readonly] 22L, 418C
                                                                                        A11
```

c언어로 구성되어있다.

파일 디스크립터란 시스템으로부터 할당받은 파일을 대표하는 0이 아닌 정수의 값이다.

프로세스에서 실행되는 파일들의 목록을 관리해주는 테이블의 인덱스 값이다.

리눅스/유닉스는 모든 장치를 파일로 관리하는데, 일반 파일과 내부/외부 모든 장치도 파일로 취급한다. 이 파일을 관리하는 것이 파일 디스크립터라고 부른다.

프로세스마다 0,1,2번으로는 사전 배정이 되어있어, 하나의 파일을 생성하게 되면 3번부터 시작하여 파일 디스크립터가 부여된다.

코드의 ato(argc [1])의 인자는 int로 밚나되어 0x1234와 뺄셈을 하여 fd변수에 값이 들어간다. 그 후 read 함수가 실행된다. read함수에서는 (int fd, void *buf, size_t nbytes) 인자이기 때문에 순서대로 파일 디스크립터, 저장할 버퍼의 포인터, 버퍼의 바이트 길이이다.

16진수 0x1234를 10진수로 변환하면 4660이 나오기 때문에 인자로 4660을 넘겨주면 입력이 가능해진다.

코드를 더 살펴보면 strcmp 함수를 통해 문자열을 비교한다. 같으면 0, 다르면 1을 반환한다. 하지만 strcmp 앞에 !이 붙어 부정이 돼버려 반대로 반환한다. 즉 LETMEWNIN\n과 buf를 비교하여 같으면 조건 달성을 하여 flag에 접근할 수 있다.

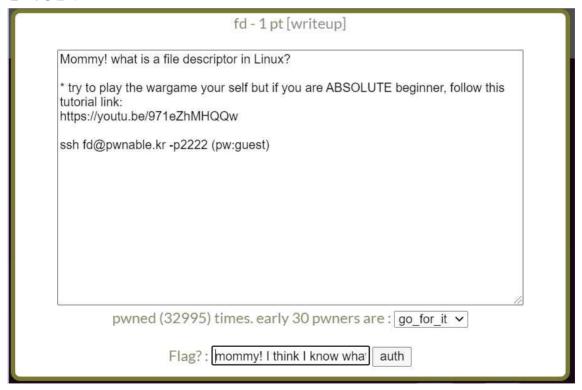
```
- to use peda, issue `source /usr/share/peda/peda.py` in gdb terminal
You have mail.
Last login: Fri Sep 24 09:53:24 2021 from 115.23.208.204
fd@pwnable:~$ vim fd.c
fd@pwnable:~$ ./fd 4660
LETMEWIN
good job :)
mommy! I think I know what a file descriptor is!!
fd@pwnable:~$
```

:q를 작성하여 나온 후,

./fd 4660

LETMEWIN

을 작성한다.



mommy! I think I know what a file descriptor is!!를 작성하면 끝!!