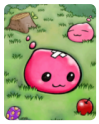
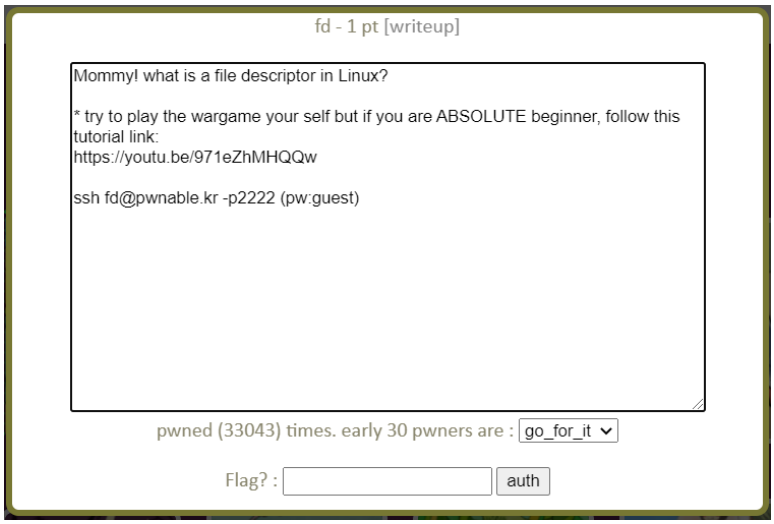


fd



먼저 문제를 풀기 위해서 SSH로 접속한다.

```
kej@kej-VirtualBox:~/바탕화면$ ssh fd@pwnable.kr -p2222
fd@pwnable.kr's password:
 ____ __    __  ____    ____  ____   _        ___      __ _  ____
|    \|  |__|  ||    \  /    ||    \ | |      /  _]    |  |/ ]|    \
|  o  )  |  |  ||  _  ||  o  ||  o  )| |     /  [_     |  ' / |  D  )
|   _/|  |  |  ||  |  ||     ||     || |___ |    _]    |    \ |    /
|  |  |  `  '  ||  |  ||  _  ||  O  ||     ||   [_  __ |     \|    \
|  |   \      / |  |  ||  |  ||     ||     ||     ||  ||  .  ||  .  \
|__|    \_/\_/  |__|__||__|__||_____||_____||_____||__||__|\_||__|\_|

- Site admin : daehee87@gatech.edu
- IRC : irc.netgarage.org:6667 / #pwnable.kr
- Simply type "irssi" command to join IRC now
- files under /tmp can be erased anytime. make your directory under /tmp
- to use peda, issue `source /usr/share/peda/peda.py` in gdb terminal
You have mail.
Last login: Thu Sep 30 06:40:48 2021 from 163.152.126.126
fd@pwnable:~$
```

```
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
```

디렉토리에는 fd, fd.c, flag 파일이 있다.

flag파일의 other는 아무 권한도 없기 때문에 flag파일을 직접 건드리는 게 아닐 것 같다.

fd.c 파일을 열어 소스코드를 확인해보면

```c
#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;

}
```

[[main 함수의 매개변수]]

- int argc - main 함수에 전달되는 인자의 갯수
- char* argv[] - main 함수에 전달되는 실질적인 인자 문자열
  (첫번째 문자열은 프로그램 실행경로로 항상 고정)

[[파일 디스크립터]]

- 0 (STDIN_FILENO) - 표준 입력(stdin)
- 1 (STDOUT_FILENO) - 표준 출력(stdout)
- 2 (STDERR_FILENO) - 표준 에러(stderr)

[[기타 함수]]

- atoi() - 문자열 → 정수
- read() - 파일 내용 읽기
  (매개변수: 파일 디스크립터, 파일을 읽어 들일 버퍼, 버퍼 크기)
- strcmp() - 문자열 비교
- system() - 실행 쉘인 /bin/sh -c 문자열을 호출하여 문자열에 지정된 명령어를 실행

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char buf[32];
int main(int argc, char* argv[], char* envp[]){
        if(argc<2){ // 인자가 2개 미만이면 아래 문장을 출력해주고 끝난다
                printf("pass argv[1] a number\n");
                return 0;
        }
        int fd = atoi( argv[1] ) - 0x1234; // 파일 디스크립트 값을 정해준다
                                           // 인자로 준 값을 정수로 변한 후 0x1234를 빼준다
        int len = 0;
        len = read(fd, buf, 32); // fd에 0이 들어가면 buf값을 표준 입력으로 받을 수 있다
        if(!strcmp("LETMEWIN\n", buf)){ // LETMEWIN와 buf 내용이 같으면 쉘을 실행한다
                printf("good job :)\n");
                system("/bin/cat flag");
                exit(0);
        }
        printf("learn about Linux file IO\n");
        return 0;

}
```

flag 파일을 실행해주기 위해

fd를 0으로 만들기 위해 argv[1]의 값을 0x1234을 10진수로 나타낸 4660으로 넣어준다.

```
fd@pwnable:~$ ./fd 4660
```

그러면 입력을 받을 수 있고, LETMEWIN값을 넣어주면

```
fd@pwnable:~$ ./fd 4660
LETMEWIN
good job :)
mommy! I think I know what a file descriptor is!!
```

flag를 얻을 수 있다.