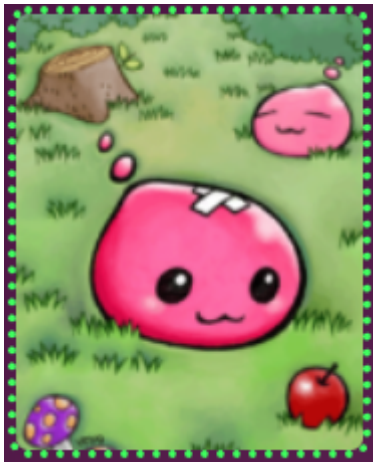
Pwnable.kr fd 풀이

리버싱 문제풀이 / Wonlf / 2022. 5. 5. 23:51



OpenSSH SSH client

```
#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;

}
                                        1,1           All
```

ssh에 접속해 보이는 c파일을 열어보니 이런 코드로 짜여져 있다.

main함수의 인자값으로 들어있는 것들의 용도로는,

argc 터미널에서 파일을 실행할 때, 뒤에 붙는 옵션의 개수 ex) a.exe -o -p 등 argc = 2

argc	터미널에서 파일을 실행할 때, 뒤에 붙는 옵션의 개수 ex) a.exe -o -p 등 argc = 2
argv	옵션의 문자열을 보여줌 하지만 argv[0]에는 파일명이 들어가있고, argv[1]부터 옵션이 들어간다. ex) a.exe -o -p 등 argv[1] = “-o”
envp	환경변수의 대한 정보가 들어있다.

그러므로 if문을 통과하려면 옵션이 하나만 있어야 한다. argv[0]은 프로그램 이름이니까.

이렇게 if문을 통과하고 atoi함수로 오게 되는데 atoi함수는 문자열을 정수로 바꿔주는 함수이다.

ex) “0x1234” ⇒ 0x1234 로 변환. fd변수에는 어떠한 정수로 초기화 되겠네요.

read함수로 넘어가서 살펴보면, read 함수의 인자값으로는,

read(**int fd, void *buf, size_t nbytes**);

int fd 읽을 파일의 파일 디스크립터

void *buf	읽어들인 데이터를 저장할 버퍼 (배열)
size_t nbytes	읽어들일 데이터의 최대 길이 (buf의 길이보다 길어서는 안된다.)

이렇게 정해져 있다. 근데 여기서 파일 디스크립터란?

리눅스에서 파일 등을 읽거나 쓸 때 사용하여 표현되는 정수이다.

0 표준 입력(Standard Input)

1	표준 출력(Standard Output)
2	표준 에러 출력(Standard Error)

그럼 read함수를 통해 buf배열에 값을 입력하려면 fd는 0이 되어야 할 것이고, 0이 되려면 argv[1]의 값이 0x1234면 fd에는 0이 들어갈 것이다. 바로 입력해본다.

안된다. 왜 안되지? atoi함수는 16진수를 지원하지 않는다고 한다..

strtol계열의 함수를 이용해야한다고....

참조 https://umbum.dev/92

그래서 0x1234를 10진수로 변환하고 실행해봤다.

그럼 이렇게 입력을 받는다. 그리고, **LETMEWIN**를 입력해주고 엔터를 누르면?

이렇게 플래그가 나오게 된다.

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

LETMEWIN를 입력할 때 뒤에 붙어 있는 ₩n을 같이 쓰지 않는 이유는 당연하게도 엔터가 ₩n로 인식되기 때문에 입력하면 안된다.