[pwnable.kr WriteUp]

이태규(ltg2030)

[Toddler's Bottle] fd 1PT

문제 설명
Mommy! what is a file descriptor in Linux? ssh fd@pwnable.kr -p2222 (pw:guest)

File Descriptor : 입출력과 관계된 시스템 콜에서 읽는 작업, 쓰는 작업, 화면 입출력, 파일 입출력 등을 선택하는 것.
이 문제에서 사용된 read 시스템 콜은 read(int fd, void *buf, size_t count)의 형태로 이루어져있다. fd는 0, 1, 2 각각 standard input, standard output, standard error를 뜻한다. buf는 배열의 포인터이고 count는 문자열의 사이즈를 의미한다.

리눅스 기본

ls -l : 현재 디렉토리의 파일과 디렉토리를 자세히 출력한다. 각각 파일, 디렉토리 별로 권한, 소유주, 소유그룹, 사이즈 등을 표시해준다.

권한 : -drwxrwxrwx 맨 처음 d는 파일인지 디렉토리인지 구분해 주는 것이고 첫 번째 rwx는 소유주의 권한을 두 번째 rwx는 소유그룹의 권한을 마지막 rwx는 그 이외의 권한을 나타내는 것이다.

SetUID, SetGID : 첫 번째와 두 번째의 x가 s인 경우 어떤 사용자가 이 파일을 실행하는 동안 그에 해당하는 권한을 갖는 것이다.

문제 풀이(이론)

[pwnable.kr WriteUp]

이태규(ltg2030)

1. 우선 Xshell을 통해 fd@pwnable.kr -p2222에 접속한 뒤 ls -l 명령어를 통해 어떤 파일들이 있는지 확인했다.

2.
-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
이런 3가지 파일이 있는 것을 알 수 있다. 우리는 지금 fd라는 사용자를 사용하고 있고 이는 root 그룹이 아니기 때문에 flag를 직접 열어볼 수는 없고 SetUID가 적용된 fd 프로그램을 이용하여 이를 열어야 하는 것을 추측할 수 있다.

3. fd.c를 열어 fd 프로그램이 어떻게 동작하는지 알아보자. cat fd.c 명령어를 사용하여 fd.c의 내용을 확인한 결과 shell에서 프로그램을 실행할 때 파라미터로 숫자를 입력받아 그 수에서 0x1234를 뺀 뒤 그것을 fd로하여 read함수를 실행시키는 것을 알 수 있다. 그 뒤로 read 함수에서 입력을 받아 그것과 "LETMEWIN\n"이라는 문자열과 비교하여 같은 경우 /bin/cat flag를 실행하는 것을 알 수 있다.

4. 따라서 read함수에서는 입력을 받아야하므로 처음 프로그램을 실행할 때 넘겨주는 파리미터 값으로는 16진수로 1234 즉 10진수로 4660값을 넘겨주어야 한다. 그렇게 fd가 0인 read 시스템 콜을 실행시켜 buf라는 배열에 최대 32비트를 입력을 받는 함수를 실행시킨다. 여기에 LETMEWIN을 입력하고 엔터를 치면 buf에는 "LETMEWIN\n"가 들어가게 되고 strcmp함수를 통해 if문의 조건이 만족되고 원하는 flag를 얻을 수 있다.

문제 풀이(실습)

```
fd@ubuntu:~$ ls -l
total 16
-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
fd@ubuntu:~$ cat fd.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;

}

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

good job :)\n 이 출력된 뒤 flag를 cat 하기 때문에 mommy! I think I know what a file descriptor is!! 가 flag 내용임을 알 수 있다.

실행화면