[pwnable.kr WriteUp]

이태규(ltg2030)

[Toddler's Bottle] mistake 1PT

문제 설명

We all make mistakes, let's move on.
(don't take this too seriously, no fancy hacking skill is required at all)

This task is based on real event
Thanks to dhmonkey

hint : operator priority

ssh mistake@pwnable.kr -p2222 (pw:guest)

연산자 우선순위 :

1	() [] -> . ::	Grouping, scope, array/member access
2	! ~ - + * & sizeof *type cast* ++x --x	(most) unary operations, sizeof and type casts
3	* / %	Multiplication, division, modulo
4	+ -	Addition and subtraction
5	<< >>	Bitwise shift left and right
6	< <= > >=	Comparisons: less-than, ...
7	== !=	Comparisons: equal and not equal
8	&	Bitwise AND
9	^	Bitwise exclusive OR
10	|	Bitwise inclusive (normal) OR
11	&&	Logical AND
12	||	Logical OR
13	?:	Conditional expression (ternary operator)
14	= += -= *= /= %= &= |= ^= <<= >>=	Assignment operators
15	,	Comma operator

(C언어의 연산자 우선순위)

우선순위에서 =이 < 연산보다 우선순위가 낮은 것을 알 수 있다.
따라서 소스의 if(fd=open("/home/mistake/password",O_RDONLY,0400) < 0) 구문에서 open 함수의 리턴 값이 fd가 되는 것이 아닌 (fd<0)의 논리 값이 fd가 된다. 따라서 open이 제대로 실행된 경우 리턴 값이 0보다 작은 것은 false 이므로 fd=0이 되고 조건문은 실행되지 않는다.

따라서 그 다음에 나오게 되는 read(fd,~~) 라는 표현식이 /home/mistake/password 라는 파일의 내용을 읽는 것이 아니라 1번 문제에서 사용했던 file descriptor를 생각해 보면 표준 입력을 받는다는 것을 알 수 있다.

xor : bit연산의 일종으로서 a ^b로 표현하며 a, b가 같으면 0 다르면 1의 값을 갖는 논리 연산이다.

문제 풀이(이론)

[pwnable.kr WriteUp]

이태규(ltg2030)

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

2.
-r-------- 1 mistake2 root 51 Jul 29 2014 flag
-r-sr-x--- 1 mistake2 mistake 8934 Aug 1 2014 mistake
-rw-r--r-- 1 root root 792 Aug 1 2014 mistake.c
-r-------- 1 mistake2 root 10 Jul 29 2014 password
이런 4가지 파일이 있는 것을 알 수 있다. 우리는 지금 mistake라는 사용자를 사용하고 있고 이는 root 그룹이 아니기 때문에 flag를 직접 열어볼 수는 없고 SetUID가 적용된 mistake 프로그램을 이용하여 이를 열어야 하는 것을 추측할 수 있다.

3. mistake.c를 열어 mistake 프로그램이 어떻게 동작하는지 알아보자. cat mistake.c 명령어를 사용하여 mistake.c의 내용을 확인하면 위의 문제 풀이(이론)에서 얘기 했듯이 buf에 내가 표준입력으로 10bytes를 입력하고 buf2에도 내가 표준입력으로 10bytes를 입력하게 된다. buf[i]^1 = buf2[i]가 되게 입력을 주면 된다.

4. buf에 “bbbbbbbbbb”를 입력해주면 ‘b’ = 98 = 0b1100010 이므로 ‘b’ ^ 1 = 0b1100011 = 99 = ‘c’ 이므로 buf2에 “ccccccccc”를 입력하면 flag를 볼 수 있다.

문제 풀이(실습)

```
$ ls -l
total 24
-r-------- 1 mistake2 root      51 Jul 29  2014 flag
-r-sr-x--- 1 mistake2 mistake 8934 Aug  1  2014 mistake
-rw-r--r-- 1 root     root     792 Aug  1  2014 mistake.c
-r-------- 1 mistake2 root      10 Jul 29  2014 password
$ ./mistake
do not bruteforce...
bbbbbbbbbb
input password : ccccccccc
Password OK
Mommy, the operator priority always confuses me :(
```

Password OK\n 이 출력된 뒤 flag를 cat 하기 때문에 Mommy, the operator priority always confuses me :(가 flag 내용임을 알 수 있다.

실행화면