[DreamHack] off_by_one_001

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>

void alarm_handler()
{
    puts("TIME OUT");
    exit(-1);
}

void initialize()
{
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    signal(SIGALRM, alarm_handler);
    alarm(30);
}

void read_str(char *ptr, int size)
{
    int len;
    len = read(0, ptr, size);
    printf("%d", len);
    ptr[len] = '₩0';
}
```

```
void get_shell()
{
    system("/bin/sh");
}

int main()
{
    char name[20];
    int age = 1;

    initialize();

    printf("Name: ");
    read_str(name, 20);

    printf("Are you baby?");

    if (age == 0)
    {
        get_shell();
    }
    else
    {
        printf("Ok, chance: ₩n");
        read(0, name, 20);
    }

    return 0;
}
```

문제에서 주어진 코드는 위와 같다. Off by one 취약점을 이용하면 되는데 age를 0으로 덮어씌우면 되는 것 같다. 아이다로 메모리를 분석해보면

```
char buf; // [esp+0h] [ebp-18h]
int v5; // [esp+14h] [ebp-4h]
```

name 변수는 ebp-18, age는 ebp-4에 있다.

둘 사이의 공간은 총 20byte이다.

Read_str함수를 보면 name에 20바이트만큼 문자열을 읽어오고 name[20]을 '₩0'으로 덮어쓴다. 20바이트면 0~19인데 name[20]에 ₩0을 더 넣는 것으로 off by one 취약점이 발생되고 여기서 name과 age 사이의 공간이 20바이트만큼 차이나니 name에서 1바이트 더 써진다면 age의 값을 0으로 덮어쓸 수 있다.

별다를 익스플로잇 코드 없이 그냥 20바이트의 문자열만 입력해도 get_shell을 실행시킬 수 있다.