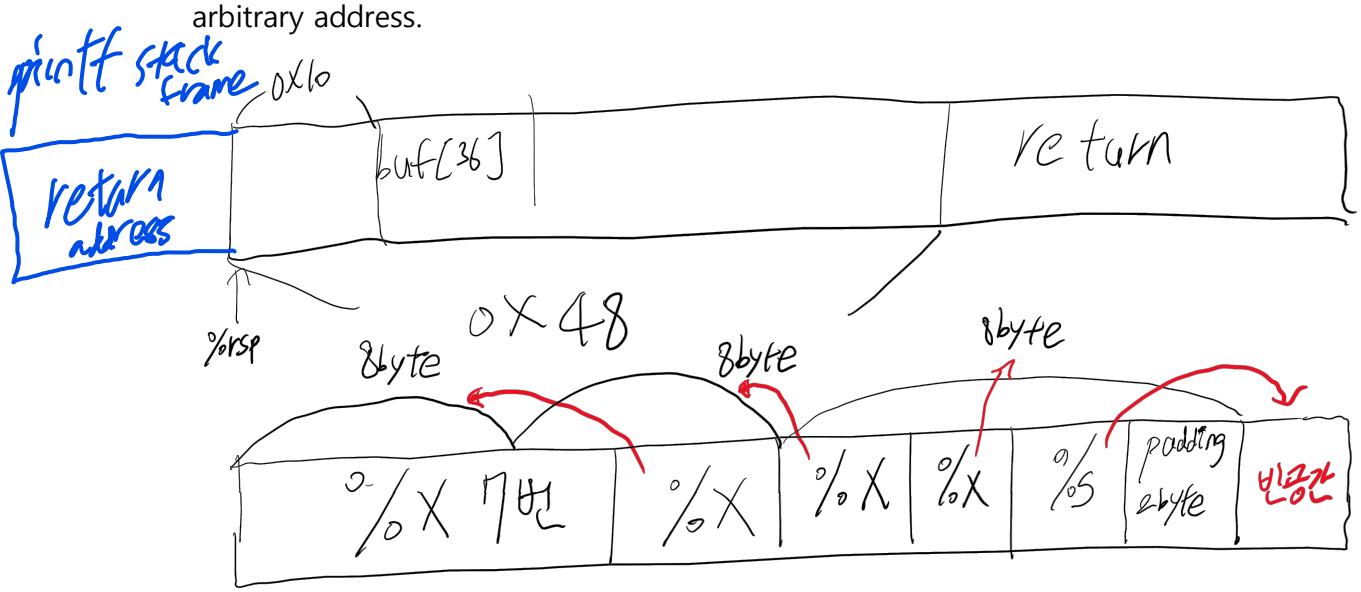
Report for 3-3

You will have to exploit a format string bug to disclose the memory content of an arbitrary address.



먼저, %x를 5번 시행하면 rsi, rdx, rcx, r8, r9에 저장된 값들이 나타난다. 이후 %x %x를 두 번 더 하면 %rsp+10의 위치에 이르게 된다. R9가 지난 시점부터는 스택에 저장된 값들이 나오게 된다. 이후 %x를 3번 더 하면 빈 공간이 위치하는 데를 발견할 수 있다. 이에 %s를 이용해 빈공간에 0x404070을 넣어 memory disclose를 일으킨다. 결국 이러한 %x를 넣는 반복과정을 거친 이유는 스택의 빈공간을 찾아 0x404070을 넣기 위함이다.

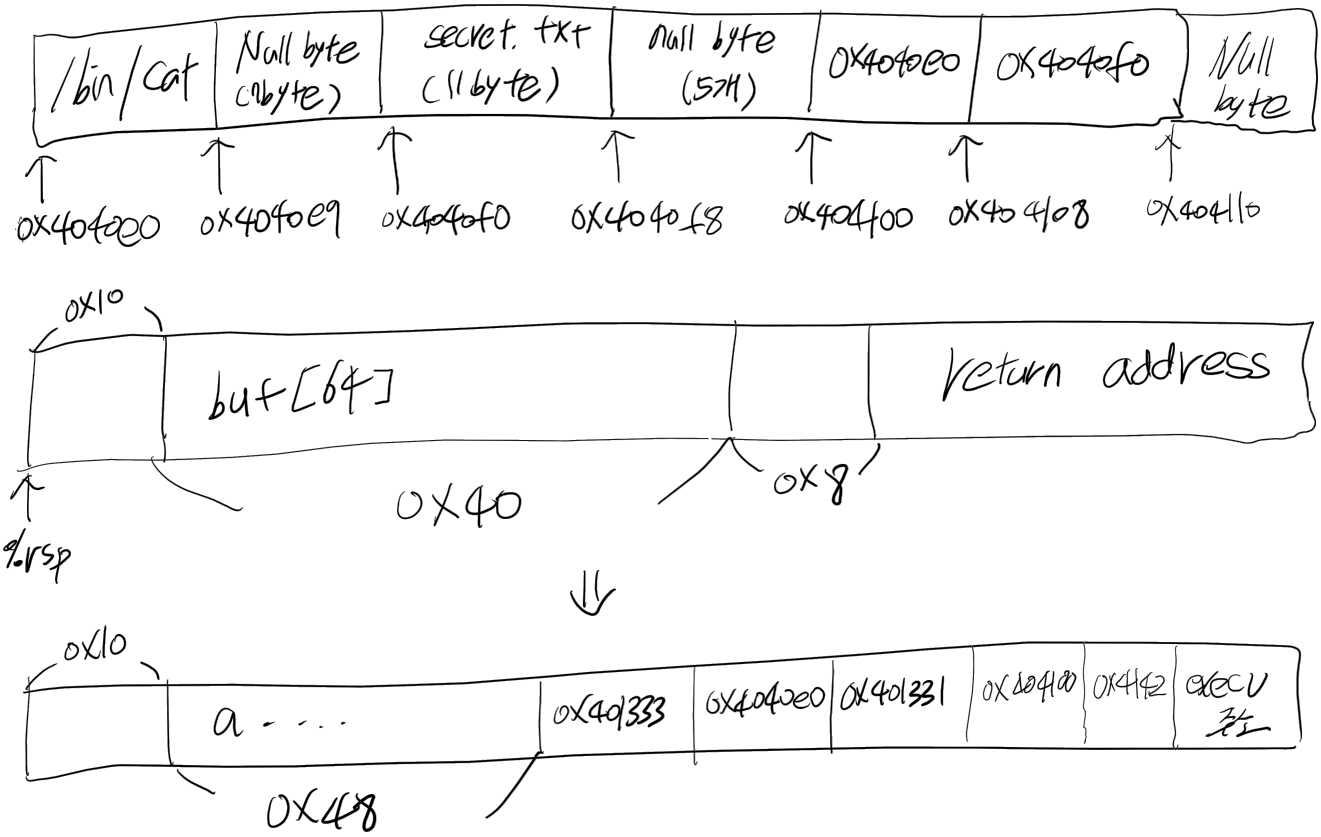
Report for 3-2

In 3-2, you must leak the addresses of libc function

먼저, i1과 i2가 음수를 검사하지 않는 다는 점과 write에서 다른 주소를 탐색할 수 있다는 점을 취약점을 보았다. 이에 이 부분을 중점으로 해킹 작업에 들어갔다. 이미

A function's GOT entry is filled in when it's called for the first time이라는 hint를 바탕으로 puts got를 gdb를 통해 축출했다. 이후 i1과 i2를 잘 조정해 library 내 puts함수 주소를 write한다. 이후 puts 주소 – puts offset + execv offset을 통해 execv 주소를 알아낸다. 이 주소를 바탕으로 입력에 들어간다.

두 번째 시행에서 input a string 입력 부분에서 1번 문제 run_cat을 참조해 rdi에 /bin/cat을 넣고, rsi에 더블 포인터 배열을 넣을 수 있게 만들었다.



더블 포인터를 이용하기 위해 /bin/cat의 주소를 0x4040e0으로 세팅하고 /bin/cat이 들어있는 0x4040e0을 0x404100에 넣고 secret.txt가 들어있는 0x4040f0주소를 0x404108에 넣는다. 이후 buffer overflow를 일으키고 rop gadget을 이용해 rdi에 /bin/cat을 넣고 더블 포인터 배열로 정의된 주소 0x404100을 불러와 rsi에 저장하고 이 두 argument를 execv에 보내 secret.txt를 출력한다. 결국 중요한 점은 더블포인터를 만들어서 rsi에 넣는 게 이번 3-2문제의 핵심이었다.

이때 Gadget 주소 0x401333에서는 rdi값을 저장하는 역할을 하고 gadget 0x401331에서는 rsi에 값을 저장하는 역할을 한다. R15는 사용되지 않기 때문에 임의의 값을 넣으면 된다.