보고서

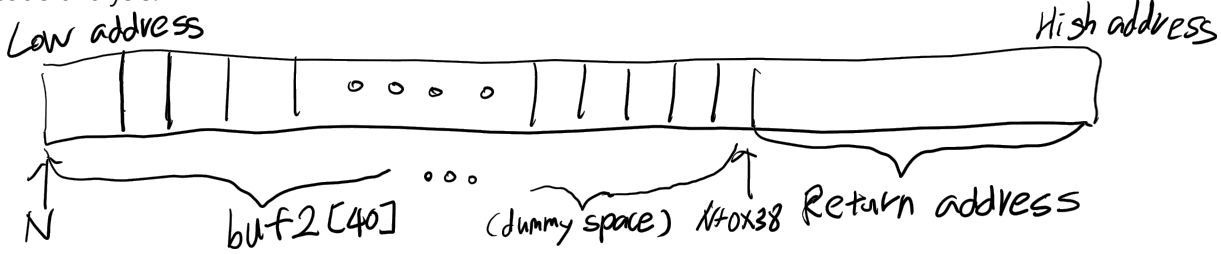
전공: 컴퓨터공학과 학년: 4학년 학번: 20212021 이름: 원대호

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)함수에서 입력 값을 받는 곳에서 일어난다. 해당 scanf 함수는 0x401279주소에서 실행된다.

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?

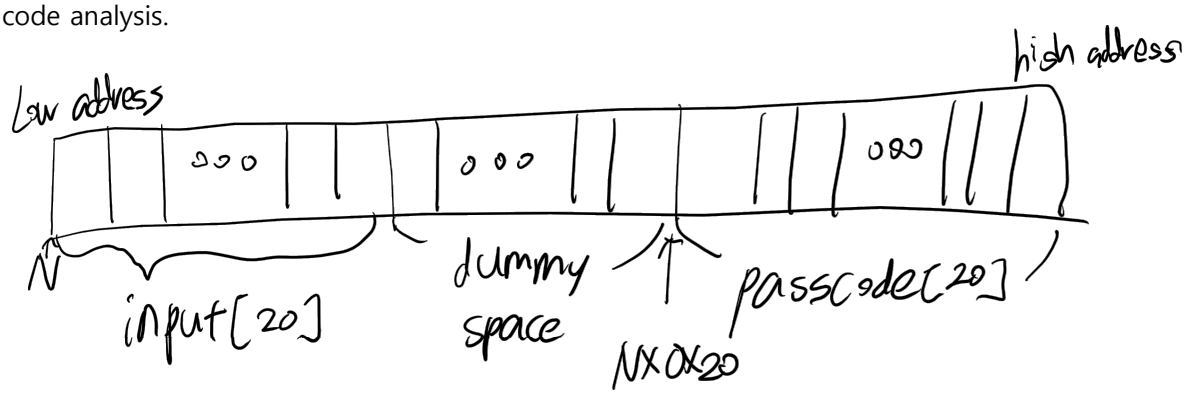
Scanf는 입력 값에 대한 길이 제한이 없기 때문에, 40바이트의 buf2배열을 초과하는 입력이 가능하다. 이로 인해 스택 메모리에서 buf2[40]을 넘어 return address주소까지 덮어쓸 수 있다. Buf2[40]이 stack에서 %rsp+0x30에 위치하고 있다. 이에 0x38(0x68-0x30)만큼 문자를 채우고 return address에 print_secret()함수의 시작 주소를 채웠다.

2-3.

Q. In source code, at which line does buffer overflow occur? What is the address of the corresponding assembly instruction?

버퍼 오버플로우는 scanf("%s", input);에서 발생한다. 해당 scanf() 호출은 0x40145f 주소에서 실행된다. scanf에서 입력 값에 대한 길이 제한이 없기 때문에, 20바이트의 input 배열을 초과하는 입력이 가능하다. 이로 인해 스택 메모리에서 input 배열을 넘어 passcode 배열까지 덮어쓰는 buffer overflow가 발생한다. 이후 0x401498 주소에서 strcmp() 함수가 호출된다. 이 함수는 오버플로우로 인해 조작된 input과 passcode 배열의 내용을 비교한다. 만약 두 문자열이 동일하다고 판단돼 strcmp()의 반환 값이 0이면, 다음 명령어인 0x40149d 주소의 jne 명령어를 건너뛰고 print_secret() 함수가 호출된다.

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?

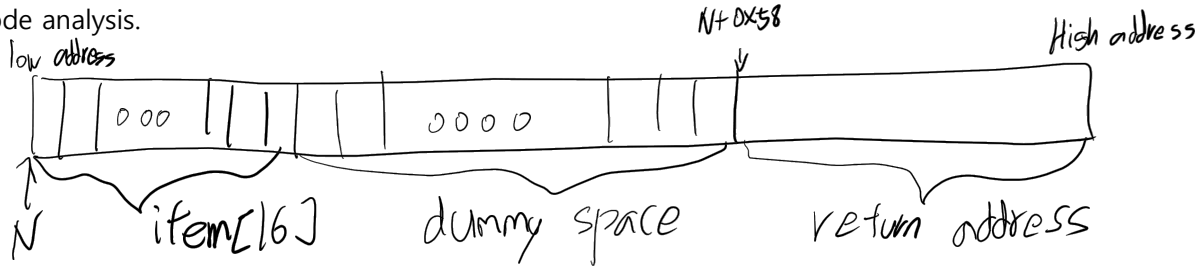
input[20]은 **%rsp + 0x10**에 저장되고, passcode[20]은 **%rsp + 0x30**에 저장됩니다. 두 배열 사이에는 32바이트의 간격이 있습니다. strcmp() 함수는 문자열을 비교할 때 널 문자를 만나면 문자열의 끝으로 인식하고 비교를 종료한다. input 배열에 'A' * 0x13 (19개의 'A')와 마지막에 널 문자(₩0)**를 삽입한다. 이렇게 하면 input 배열을 20바이트로 가득 채우며, 이후 버퍼 오버플로우를 유발해 스택의 나머지 공간과 passcode 배열을 'A' * 0x1F로 덮어쓴다. 마지막으로 passcode의 끝 부분에 널 문자를 추가해, strcmp()가 두 배열을 비교할 때 동일한 값으로 인식되도록 만든다. 이렇게 passcode와 input을 난수와 상관없이 동일하게 만들어 print_secret()함수를 동작시킨다.

2-4.

Q. In source code, at which line does buffer overflow occur? What is the address of the corresponding assembly instruction?

0x4012ee 주소에서 scanf()를 통해 사용자의 입력을 받는다. 이때 입력된 값이 유효한 범위를 벗어난 인덱스를 포함하면, 이후 배열 경계를 넘는 메모리 접근이 발생할 수 있다. 이러한 입력은 rebalance_portfolio() 함수에서 문제를 유발한다. 0x4013b9 주소에서 Withdraw ID를 입력받고, 0x4013cf 주소에서 Invest ID를 입력받으며, 0x4013e5 주소에서 Amount 값을 입력받는다. 버퍼 오버플로우는0x401402, 0x401409 주소의 명령어에서 발생한다. 여기서는 sub %eax, (%r12, %rbp, 4) 명령어가 items[src_idx]의 값에서 amount를 빼는 작업을 수행하고, 이어서 add %eax, (%r12, %rbx, 4) 명령어가 items[dst_idx]에 amount를 더하는 작업을 수행한다. 이 명령어들은 2번의 반복 과정을 통해 배열의 경계를 넘어 return address를 print_secret으로 덮어쓴다.

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?

items[i]에 값을 조작하여 리턴 주소에 직접 접근한다. 여기서 0x58 바이트는 4바이트 단위로 나누면 22가 나오므로, items[22]가 리턴 주소를 가리키게 된다. 이에 items[22]에 접근해 리턴 주소를 덮어쓸 수 있다. 이 과정을 수행하기 위해서는 총 2번의 반복 과정이 필요한다. 첫 번째 단계에서는 리턴 주소의 원래 값을 삭제해야 한다. 리턴 주소 0x4014fa를 삭제하기 위해 scanf()는 10진수 입력만 허용하므로, 0x4014fa를 10진수로 변환한 값을 amount에 할당해 원래 주소를 삭제한다. 두 번째 단계에서는 print_secret() 함수의 주소를 리턴 주소에 할당한다. 마찬가지로 print_secret()의 주소를 10진수로 변환한 값을 amount에 입력하고, 이를 사용해 items[22]에 접근해 리턴 주소를 덮어쓴다. 이러한 과정을 통해 프로그램의 리턴 주소가 print_secret() 함수의 주소로 변경된다.v