Lab2 Report

20200422 이수빈

2-1

myecho.c의 취약점은 제한된 크기의 배열에 제한되어 있지 않은 길이의 문자열을 입력받아 저장한다는 점이다. x/10xg $rsp 명령어를 통해 10번째 chunk에 return address인 0x40076f가 저장되어 있음을 알 수 있다.

따라서 9개의 chunk를 채우기 위해 72 bytes를 버퍼에 입력하고 print_secret의 시작 주소인 0x4006a6까지 보내주면 BOF를 통해 print_secret 함수를 성공적으로 실행시킬 수 있다. p.sendline(b"A" * 0x48 + b"\xa6\x06\x40")와 같은 방식으로 이를 구현할 수 있다.

2-2

guess.c의 취약점은 제한된 크기의 배열에 제한되어 있지 않은 길이의 문자열을 입력받아 저장한다는 점이다. main 함수의 assembly code를 보면, rsp + 0x10에 저장된 input과 rsp + 0x30에 저장된 passcode를 strcmp의 인자로 넘겨주는 것을 알 수 있다. 즉, input을 입력 받는 부분에서 input과 passcode 사이의 메모리 공간을 모두 채워주면 passcode의 공간을 침범해 원하는 대로 overwrite할 수 있음을 알 수 있다.

따라서 p.sendline(b"A" * 32 + b"B" * 8)와 같은 방식으로 이를 구현할 수 있고, 이후 passcode 값으로 BBBBBBBB라는 문자열을 입력하면 print_secret 함수를 성공적으로 실행시킬 수 있다.

2-3

fund.c의 취약점은 제한된 크기의 배열에 제한되어 있지 않은 크기의 정수를 입력받아 배열 인덱스로 사용하고 있다는 점이다. 배열의 시작 주소는 0x7fffffffe3c0이고, 해당 주소가 0번째 인덱스일 때 return address인 0x400d15는 22번째 인덱스 위치에 저장되어 있음을 확인하였다.

따라서 해당 메모리에 저장되어 있는 값에 적절한 값(16진수로 41f, 10진수로 1055)을 subtract하여 print_secret의 시작 주소인 0x4008f6으로 바꿔주었다.