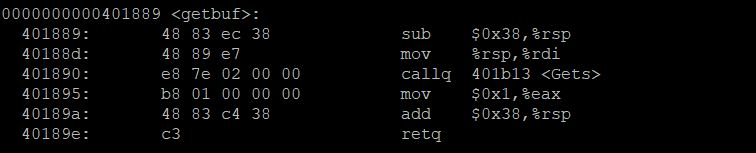
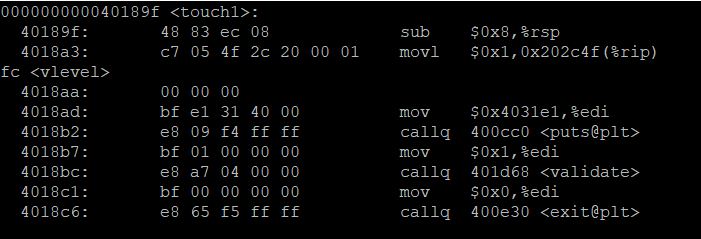
Attack Lab report (target53)

202155536 김주송

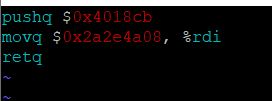
**<phase\_1>**

getbuf를 확인하면 0x38 byte의 buffer를 할당한다.

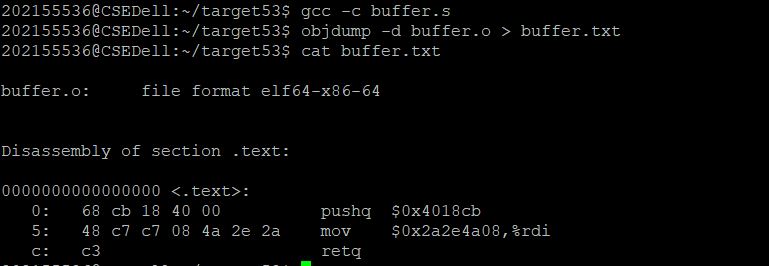
56byte의 padding을 입력한 후, touch1의 address를 입력해야 한다.

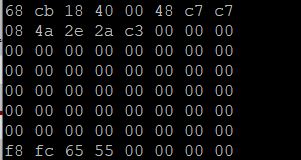
**<phase\_2>**

Pdf를 참고하면, %rdi register를 수정하고, 쿠키를 저장해야 한다.

cookie.txt에서 쿠키값을 알아내고

rdi register에 이를 저장하는 코드를 작성한 후,

gcc -c로 object 파일을 생성하고 odjump -d를 수행하여 어셈블리 코드를 알아낸다.

gdb 실행시켜 getbuf에 break를 걸어 rsp값을 알아낸다.

먼저 구한 어셈블리 코드 값을 집어넣고, rsp값과 56bytes를 채우기 위한 padding을 넣어 작성한다.

**<phase\_3>**

touch3 함수를 호출하고 쿠키를 문자열로 전달해야 한다.

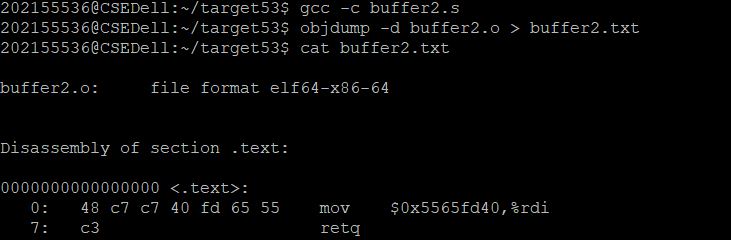
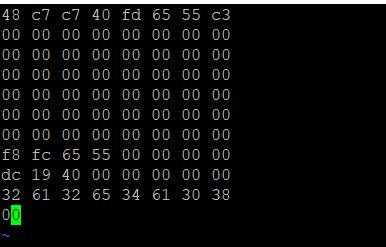
getbuf에 할당된 버퍼에 쿠키를 저장하면 hexmatch 함수가 데이터를 스택에 푸시하기 때문에 쿠키를 덮어쓸 수 있으므로 주의해야 한다.

Touch3 이후에 쿠키를 보관하기 위해 $rdi에 쿠키의 주소를 전달한다.

쿠키 전의 총 바이트는 buffer + rsp의 반환 주소에 대한 8byte + touch3에 대한 8byte

0x38+8+8 = 0x48

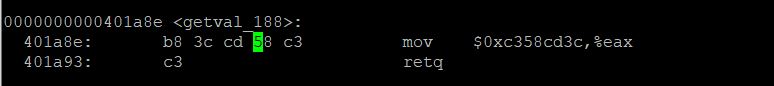
rdi에 %rsp+0x48을 해준다.

Gcc와 objdump를 이용하여 assembly code를 알아낸다.

먼저 assembly code 값을 넣고, rsp의 값과 56byte를 위한 padding을 넣은 후, touch3의 주소를 넣어주고, cookie값을 아스키코드로 변환한 string을 집어넣는다.

**<phase\_4>**

4는 2처럼 touch2를 호출하고 쿠키를 전달한다. 다른 점이 있다면 가젯을 이용하여 실행한다는 점이다.

popq %rax (58)을 찾고, movq %rax, %rdi (48 89 c7)을 찾으면

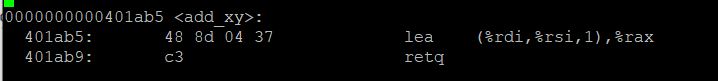
popq %rax는 0x401a91, movq %rax %rdi는 0x401a9d이다.

padding 56 bytes를 넣어주고, popq %rax address를 넣고, cookie값을 넣어준다.

그 후, mov %rax, %rdi의 address를 넣어주고 touch2의 주소를 넣어 해결한다.

**<phase\_5>**

문자열을 스택에 저장해두고, 이 문자열의 주소를 rdi register에 담아야 한다.

rsp register를 통해서 특정 시점의 stack 주소를 얻을 수 있다.

add\_xy 함수는 rdi와 rsi를 더해서 rax에 넣는 역할을 한다.

이를 수행하기 위한 과정으로

56 byte buffer를 먼저 넣어준 후,

1. movq %rsp, %rax (48 89 e0) => e0 1a 40 00 00 00 00 00 00
2. movq %rax, %rdi (48 89 c7) => 9d 1a 40 00 00 00 00 00 00
3. popq %rax (58) => 91 1a 40 00 00 00 00 00
4. offset => 48 00 00 00 00 00 00 00
5. movl %eax, %ecx (89 c1) => da 1a 40 00 00 00 00 00
6. movl %ecx, %edx (89 ca) => 3c 1b 40 00 00 00 00 00
7. movl %edx, %esi (89 d6) => 43 1b 40 00 00 00 00 00
8. add\_xy function address => b5 1a 40 00 00 00 00 00
9. movq %rax, %rdi (48 89 c7) => 9d 1a 40 00 00 00 00 00
10. touch3 address => dc 19 40 00 00 00 00 00
11. cookie string => 32 61 32 65 34 61 30 38

을 각각 넣어주면 해결된다.