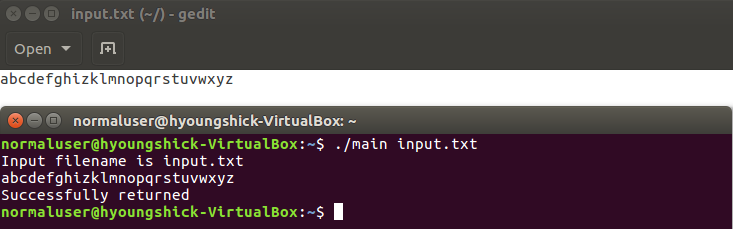
정보보호개론 Assignment 3

2016310936 우승민

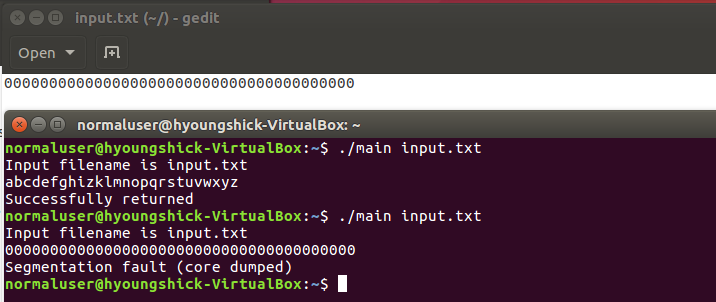
이번 과제는 Buffer Overflow를 이용하여 memory corruption attacks을 통해 shell의 root 권한을 사용할 수 있게 만드는 것입니다.

우선 과제에서 memory corruption attack에 사용될 program인 main 파일에 대해서 살펴보면

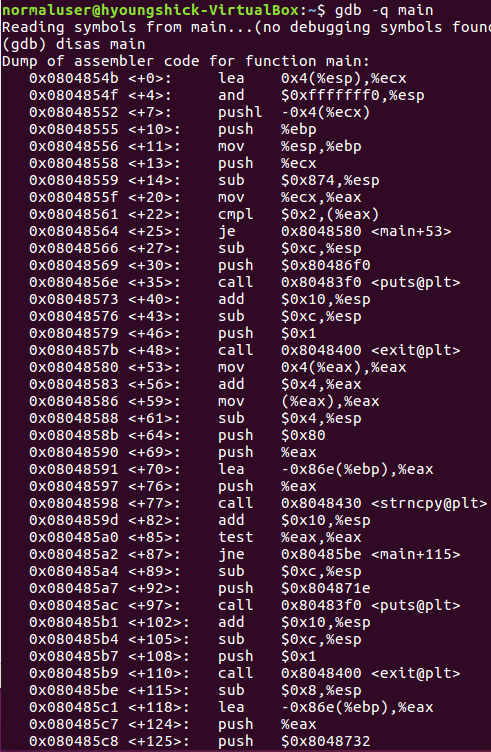


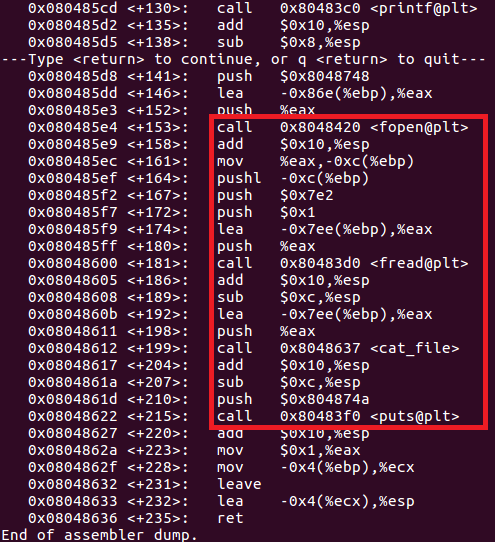
input.txt 파일을 2번째 인자로 넣어주면 input.txt 파일에 있는 string을 출력해주는 프로그램입니다.

그러나 아래와 같이 input.txt에 string의 크기를 크게 할 경우에는 segmentation fault가 발생하게 됩니다.



문제는 아마도 buffer의 크기를 초과하는 양을 input.txt에 넣어주었기 때문일 것이고, 따라서 가장 처음 해야 할 것은 main에서 사용한 buffer의 크기를 찾는 것입니다.

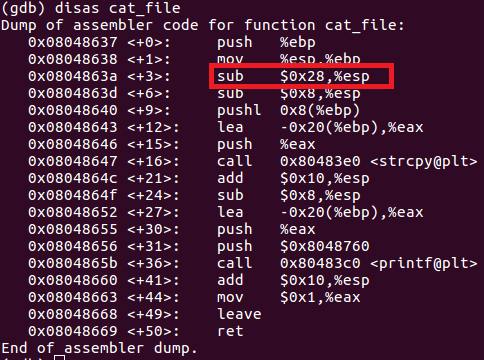
그러기 위해 main 파일을 gdb를 사용하여 보면 아래와 같이 이루어져 있습니다.

여기서 주목해야할 곳은 네모로 표시한 부분입니다.

fopen 함수로 파일을 열고, fread함수로 파일을 읽은 뒤 마지막으로 put 함수에서 출력하는 것으로 보입니다.

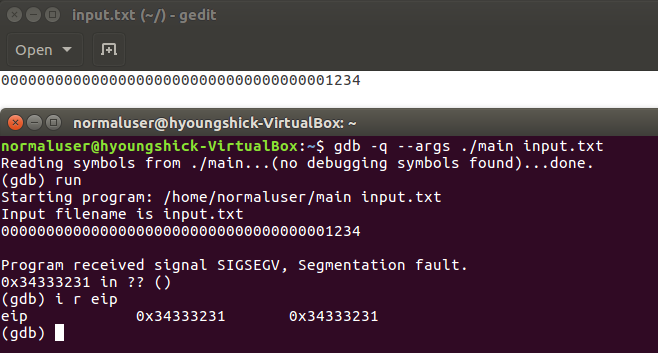
여기서 put 이전에 있는 cat\_file 함수에서 input.txt에 있는 내용을 buffer에 옮기는 것으로 추정됩니다.

cat\_file의 함수를 확인하면 아래와 같습니다.

여기서 표시한 부분인 sub instruction에서 buffer의 크기를 확인하게 됩니다.

0x28 = 40

확인을 하기 위해 input.txt 파일을 수정한 후 gdb를 통해 main파일을 실행하면



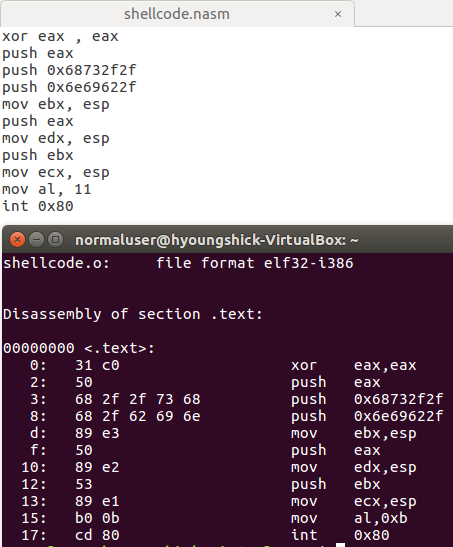
현재 0을 36번 입력한 후 1234를 input.txt에 넣어주었고 이를 gdb를 사용하여 mian을 실행한 것입니다. 0의 ascii code가 0x30 이므로 현재 eip의 값인 0x34333231은 1234로 덮어 씌어 졌다는 것을 알 수 있고, offset이 40이 맞다는 것을 확인 할 수 있습니다.

과제의 목표는 root 권한으로 shell을 실행하는 것이므로, input.txt에 들어가야 하는 내용은

**dummy 글자 36 + 이동할 주소 + nop\*? + shellcode 로 이루어져야 합니다.**

이렇게 하는 eip가 덮어 쓰여지는 부분을 이동할 주소로 만들어주고, shellcode를 실행하여 root 권한으로 shell을 실행할 수 있도록 한 것입니다. 여기서 nop의 역할은 이동할 주소의 위치에 넓게 nop을 써주면, 그 부분을 넘어가서 shellcode를 실행할 수 있기 때문입니다.

현재 과제 pdf에서 주어진 shellcode는 아래와 같이 이루어져 있으며, 이는 root 권한을 얻는 부분이 없습니다.

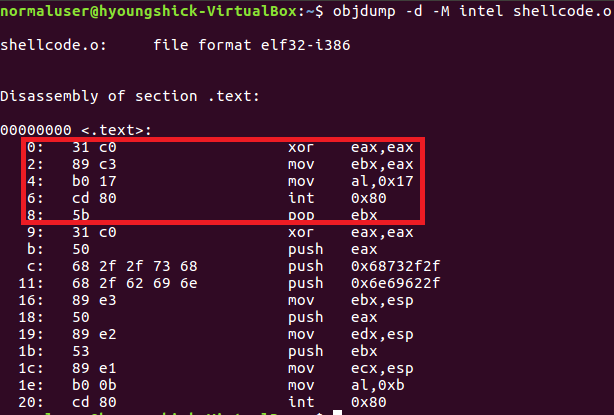


따라서 root 권한을 얻기 위해서 setuid(0)을 실행하는 부분이 추가 되어야합니다.

system call table에서 사용할 함수인 execve와 setuid의 number을 보면 아래와 같이 0x0b, 0x17인 것을 알 수 있습니다.

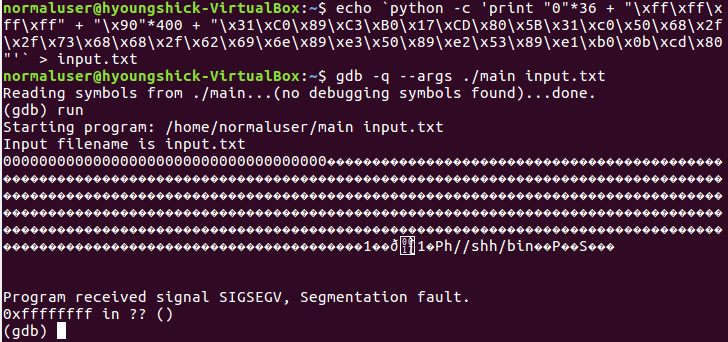


과제에서 주어진 assembly에 setuid 부분을 추가하여 제가 만든 것은 아래와 같습니다. 표시한 부분이 추가된 부분입니다.( setuid(0))

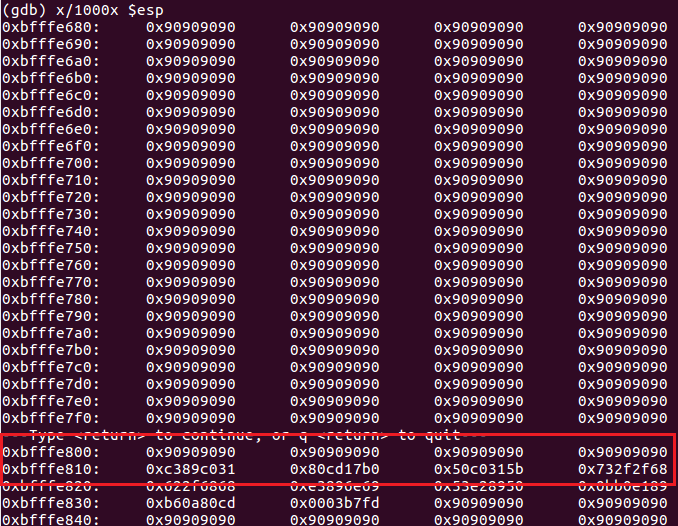


마지막으로 사용할 nop의 크기와 그에 따른 이동할 주소를 구해야 합니다. 저는 nop을 400개 사용하였습니다. echo와 python을 사용해 input.txt에

**0\*36 + FFFFFFFF + nop \* 400 + shellcode**을 입력해 준 후 gdb로 실행하면



위와 같이 나오고, 그 후 esp의 기록을 확인합니다.

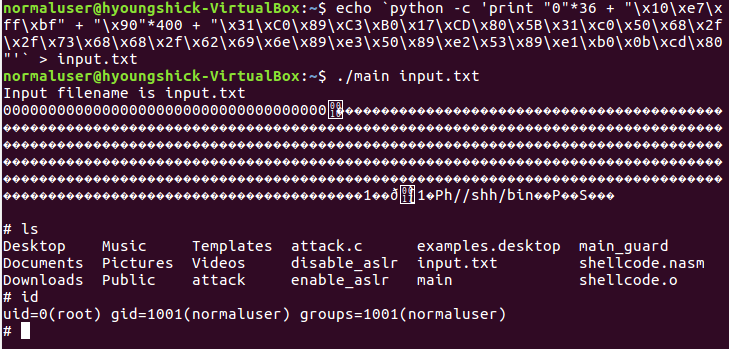


표시한 부분부터 nop 이외 값이 출력되는 것을 알 수 있습니다.

따라서 표시된 부분 위 부분에서 주소를 지정하면 됩니다.

저의 경우는 주소를 0xbfffe710 을 사용하였습니다.

이제 필요한 것을 모두 구했으므로 이를 바탕으로 input.txt에 넣어서 main을 길행 합니다.

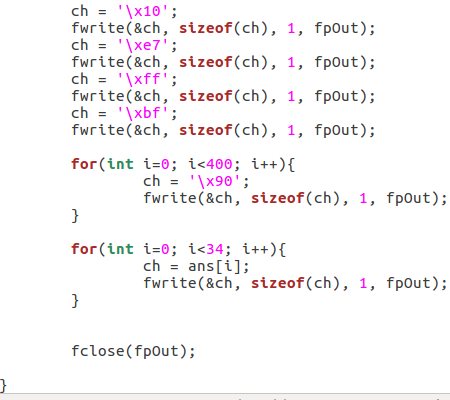


id instruction 을 통해 확인해보면, root 권한을 얻었다는 것을 확인할 수 있습니다.

이제 이를 바탕으로 attack.c code를 작성하면

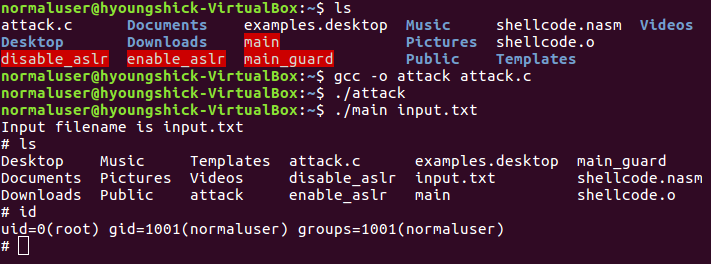


먼저 shellcode를 ans라는 char형 array에 저장하고, main문에서 “input.txt” 파일을 열어줍니다. 그 후 처음 dummy 부분인 0을 36번 반복하여 입력합니다.



그 후 주소에 해당하는 부분을 하나씩 입력해준 후 nopt을 400번 입력하고, 마지막으로 ans에 저장한 shellcode를 순서대로 파일에 입력해줍니다.

마지막으로 attack.c code를 바탕으로 memory corruption attack을 실행한 화면입니다.



저의 경우는 ASLR을 disable할 경우에만 정상적으로 실행됩니다.(주소를 지정해 주었기 때문에)