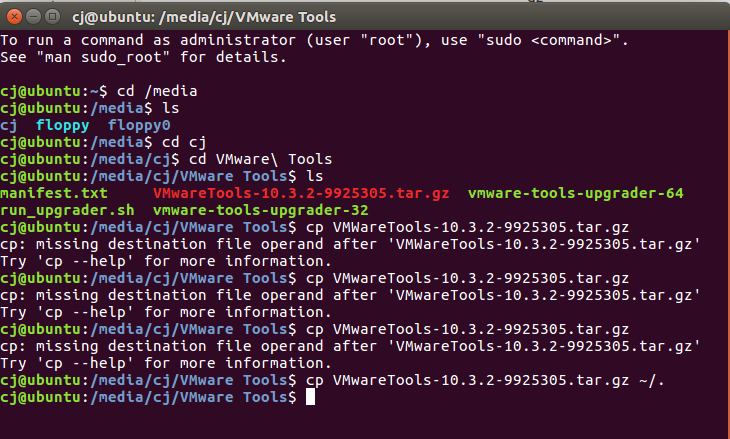
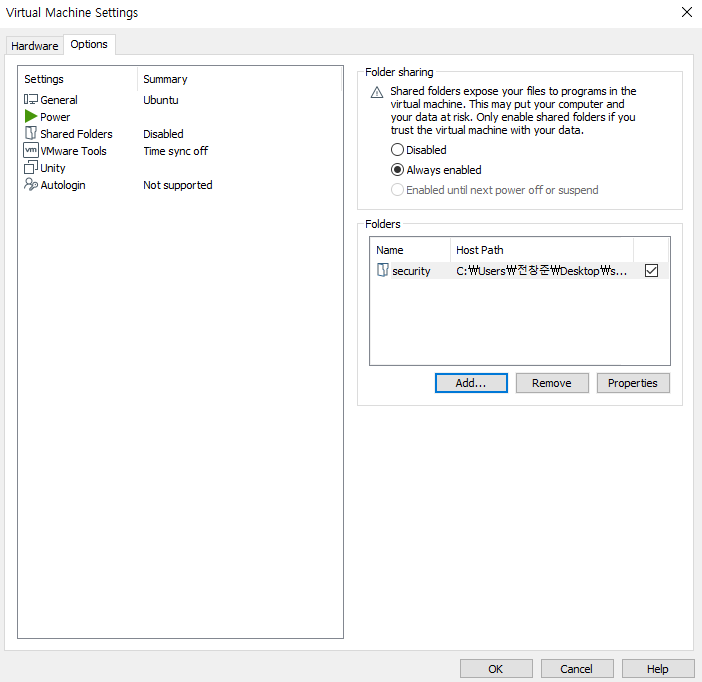
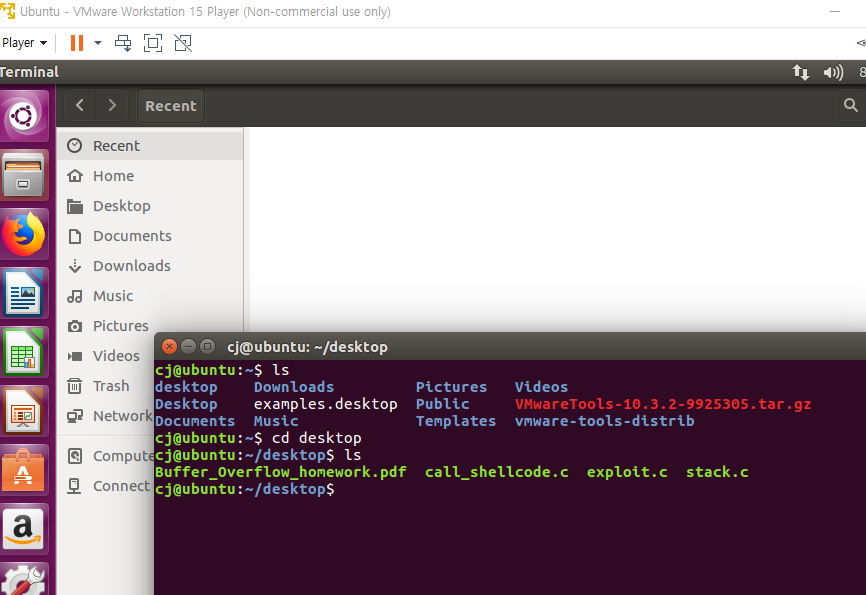
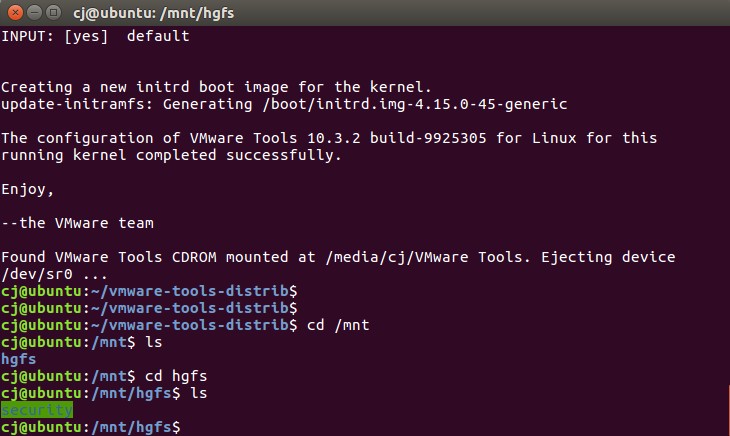
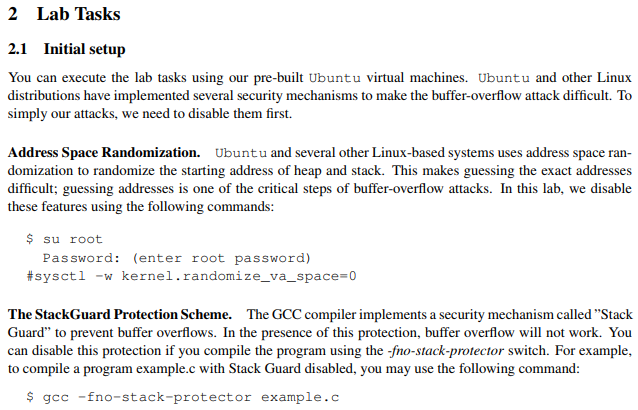


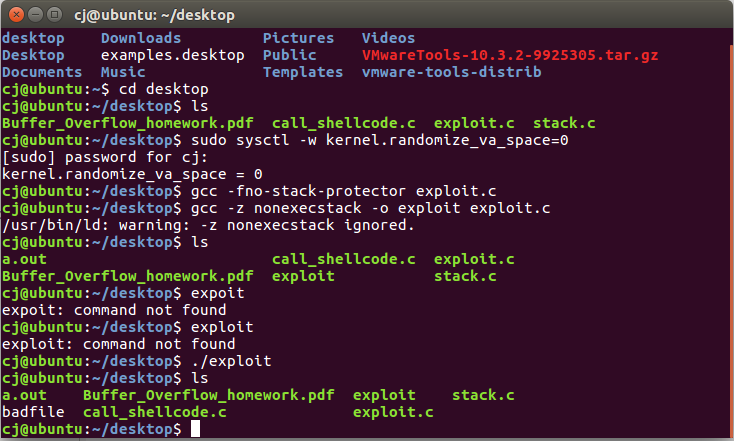
Vmware에 Ubuntu 16.04버전을 깔았습니다.





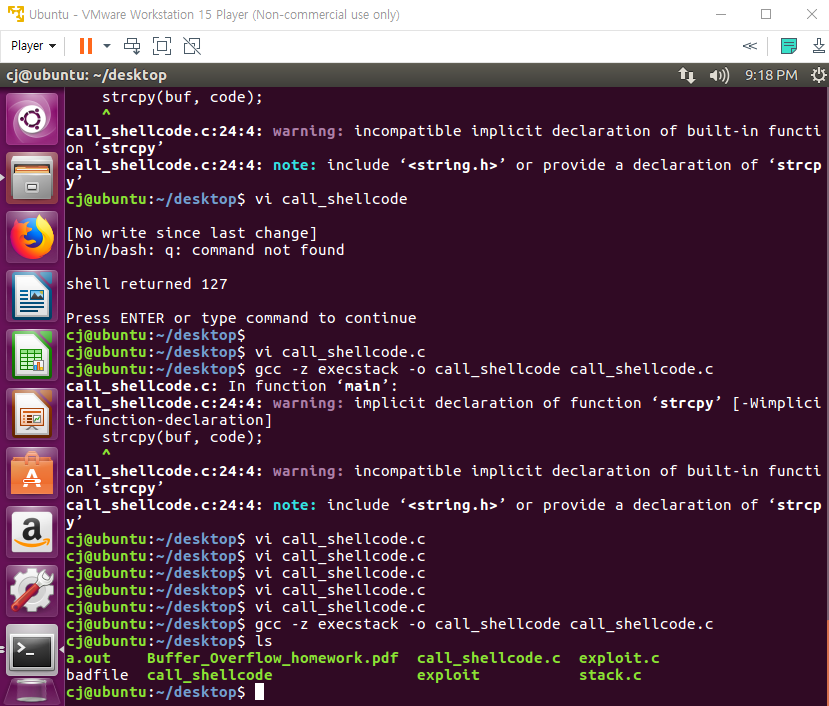
[[1]](#footnote-13593)과제를 위하여 다운로드 받은 C 파일들을 옮기는 작업을 하였습니다.



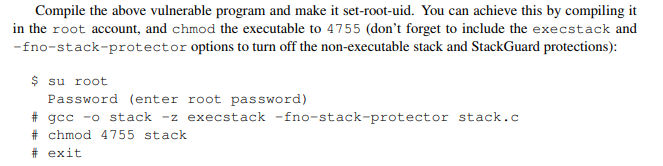


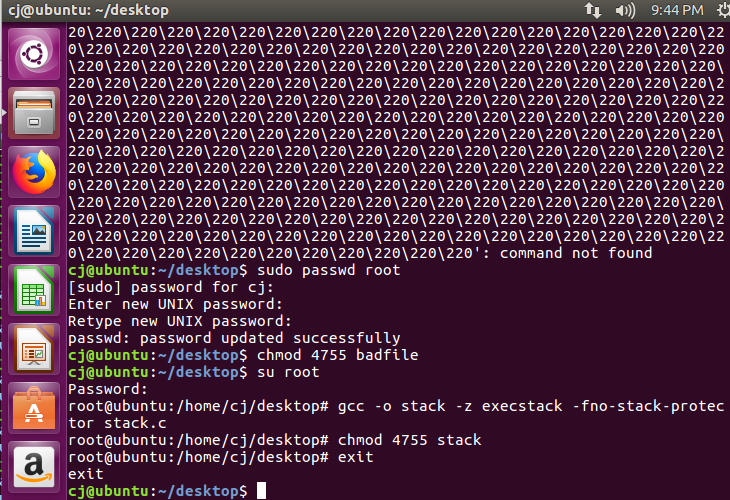
ASLR이 이미 적용되어 있기 때문에 ASLR을 해제하였고, nonexecstack 옵션으로 exploit.c 파일을 실행하였습니다.



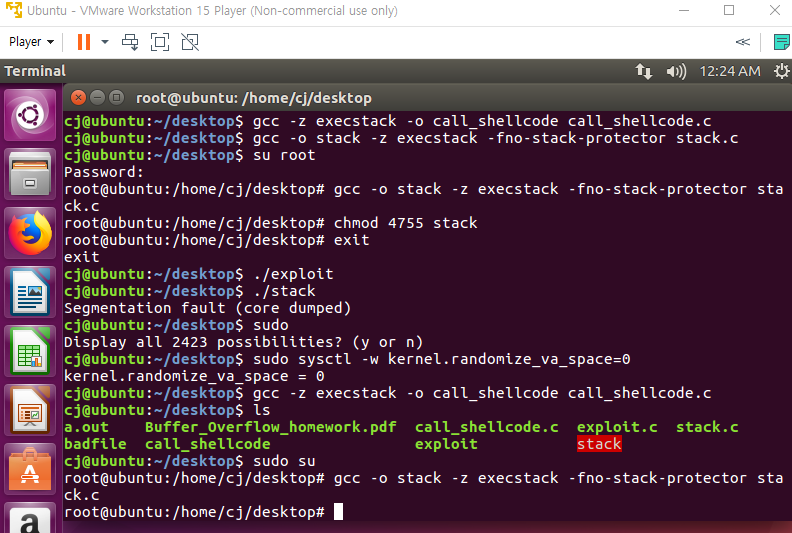


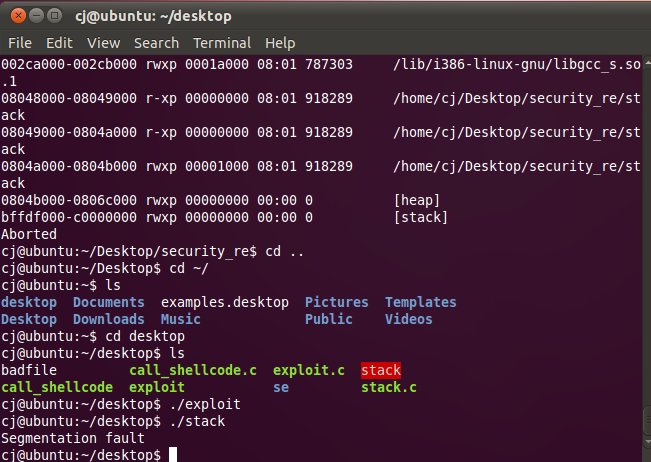
call\_shellcode.c 파일에 <string.h> 헤더를 선언하지 않아서 선언한 뒤 execstack 상태로 컴파일하였습니다.



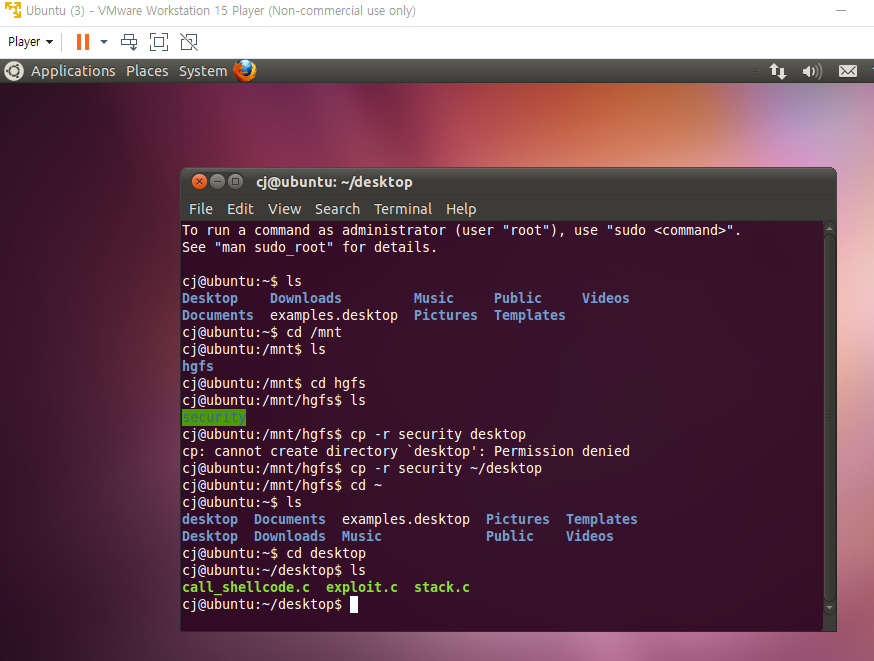


stack에 대해서 execstack 상태로 컴파일하고, stack 파일에 대한 권한 설정을 하였습니다.

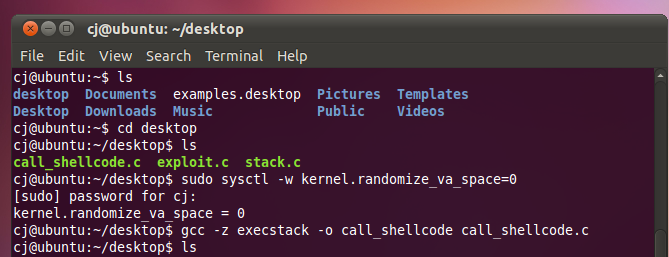




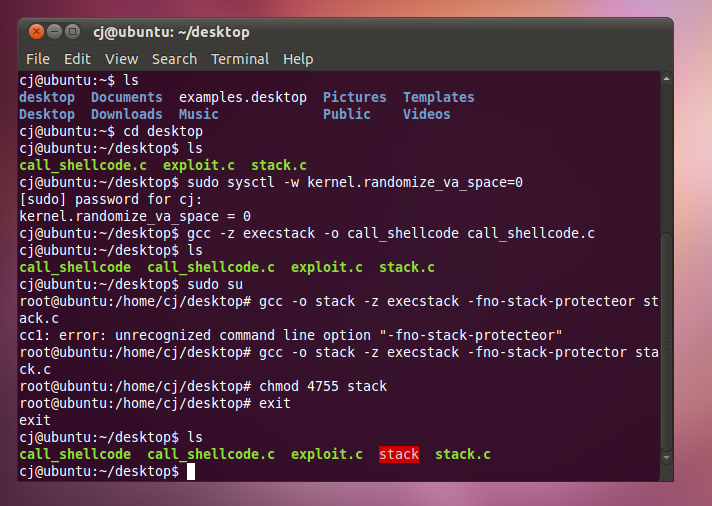
하지만 segment 오류가 계속 발생하였습니다.



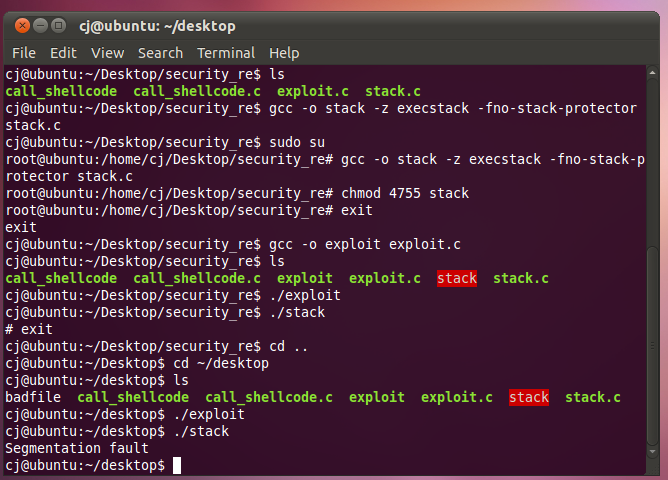
16.04버전 자체가 버전이 높다는 지적이 나와서 11.04버전으로 다운그레이드 하였습니다.



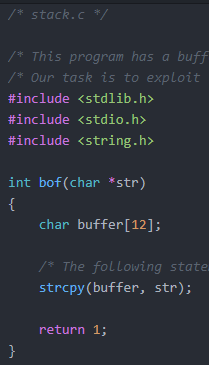
ASLR을 0로 선언하였고, call\_shellcode 파일을 stack메모리에 실행권한을 주어 컴파일하였습니다.



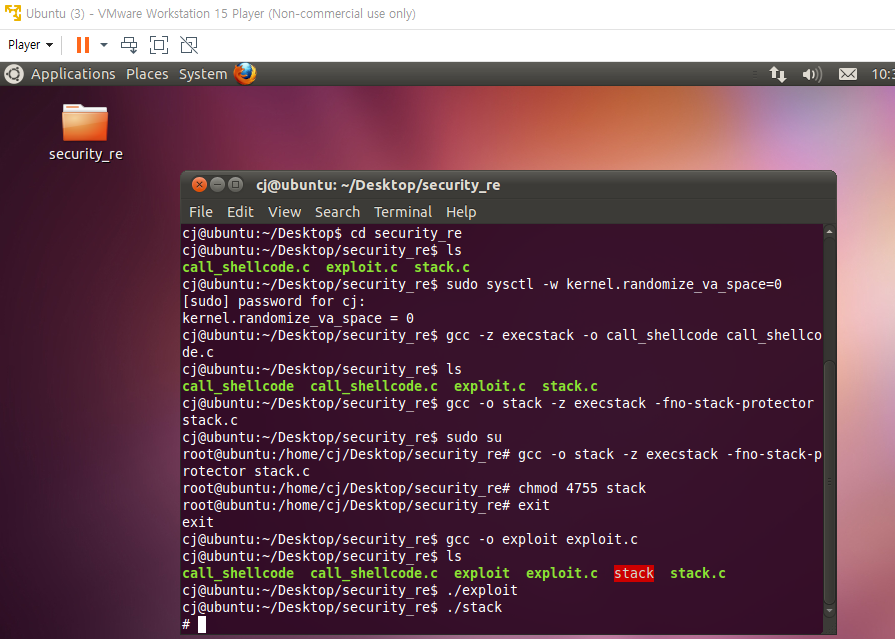
그 다음 루트 권한으로, stack 보호기법을 해제한채로 stack.c를 메모리 실행권한으로 컴파일하였습니다.



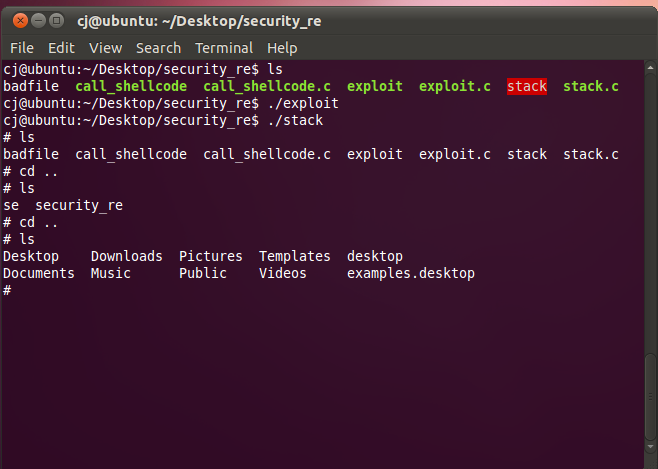
그래도 segment 오류가 나왔습니다. 그래서 왜 이런 현상이 나오는지 모든 c파일에 대해서 분석을 한 결과



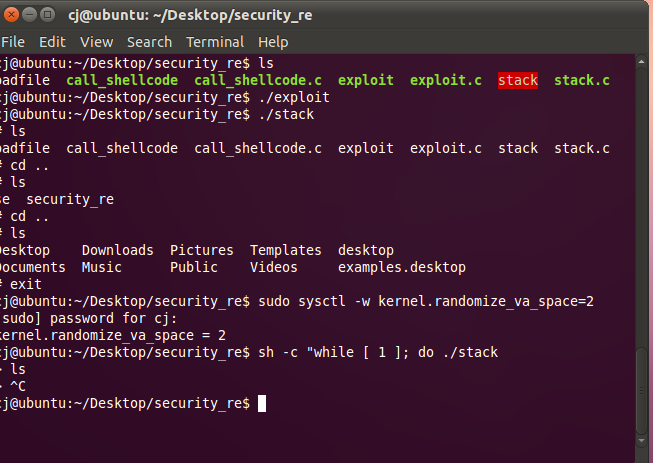
Buffer 부분의 할당 값을 12가 아닌 24로 한 것이 원인입니다.



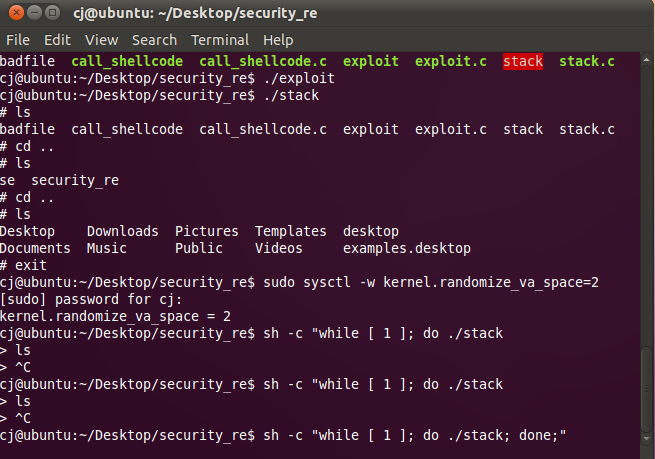
성공적으로 ROOT 권한을 탈취하였습니다.

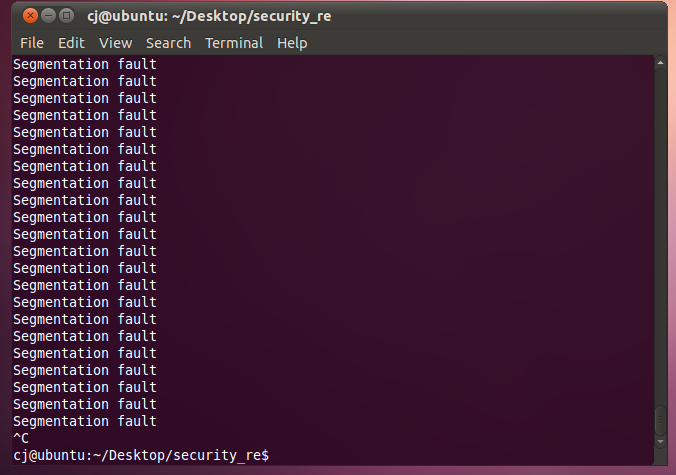


ROOT 권한이 정상적으로 작동하는지 확인하였습니다.

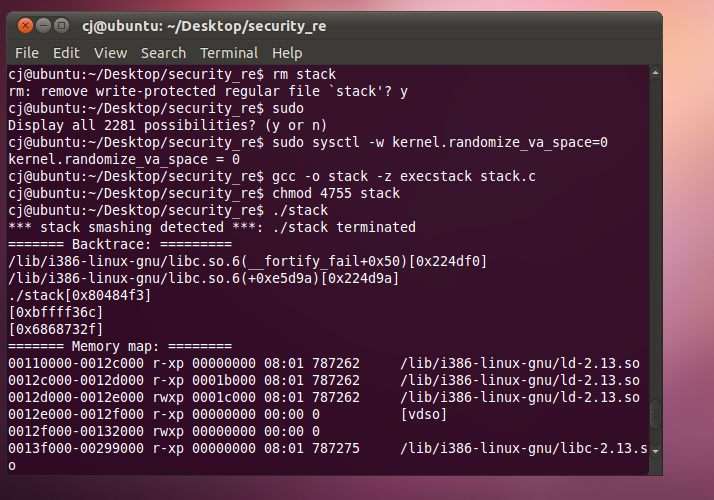


Stack 파일을 이용하여 랜덤 스택/힙/라이브러리를 활성화시키고도 공격할 수 있는지 확인해보았습니다.





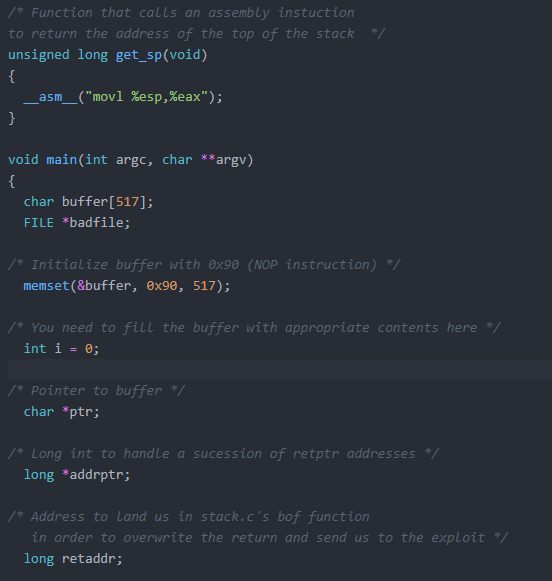
정상적으로 작동하는 것을 확인할 수 있었습니다.



Stack 파일을 제거하고 난 뒤 stack 파일에 대해서 권한 stack에 대해서 4755모드로 한 뒤에 메모리 맵을 본 것입니다. [[2]](#footnote-11447)과제를 하는데 도움이 된 자료입니다.

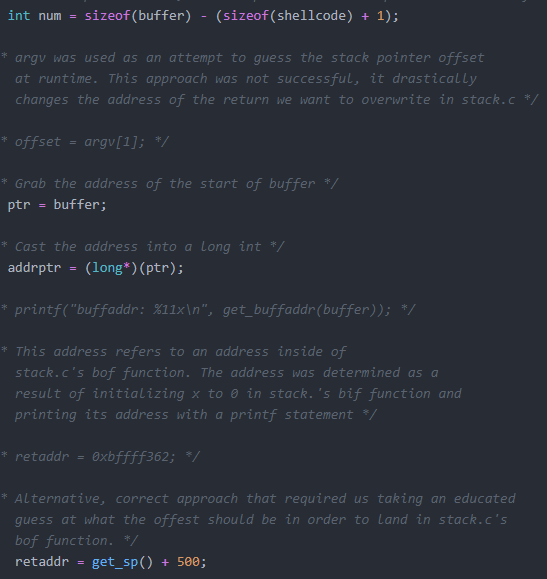
도움이 된 자료와 교수님이 주신 파일과 비교해보았을 때 buffer의 크기가 24인 것과 다른 차이가 있었습니다. 바로 exploit.c 파일의 내용이 달랐습니다.

Exploit.c파일에 대한 설명입니다.



get\_sp라는 함수를 호출 했을때, 어셈블리어로 esp의 값을 eax로 옮기는 작업을 하게 됩니다.

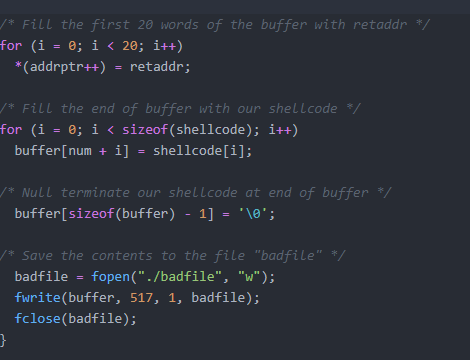
[[3]](#footnote-27339)Memset(&buffer, 0x90, 517)은 buffer가 가르키는 메모리를 517바이트만큼 0x90으로 설정합니다.



버퍼의 크기에 shellcode값을 빼고 그 값에 개행문자의 크기만큼 더한 값을 num에 대입합니다.

Addptr은 buffer의 처음을 가르키게되고,

Retaddr은 어셈블리어 명령어 값에 500바이트만큼 더한 값을 저장하게 됩니다.



그리고 첫번째 20단어를 버퍼의 ret값으로 채우게 됩니다.

그리고 buffer에 shellcode값을 채워넣게 되고, 개행문자까지 집어넣은 후 badfile에 buffer값을 저장하게 됩니다.

1. "윈도우에서 vmware로 파일 옮기기/ vmware 파일 공유." *http://blog.naver.com/*. n.d. 수정, 2019년 5월 15일 접속, http://blog.naver.com/PostView.nhn?blogId=wlgh325&logNo=221387701586&categoryNo=0&parentCategoryNo=0&viewDate=¤tPage=1&postListTopCurrentPage=1&from=postView. [↑](#footnote-ref-13593)
2. "CMPE 220 Lab2 - Buffer Overflow Vulnerability Lab," *https://www.youtube.com/*, n.d. 수정, 2019년 5월 16일 접속, https://www.youtube.com/watch?v=Hvg27trhpZ8. [↑](#footnote-ref-11447)
3. "35.3 메모리 내용을 한꺼번에 설정하기," *https://dojang.io/*, n.d. 수정, 2019년 5월 17일 접속, https://dojang.io/mod/page/view.php?id=287. [↑](#footnote-ref-27339)