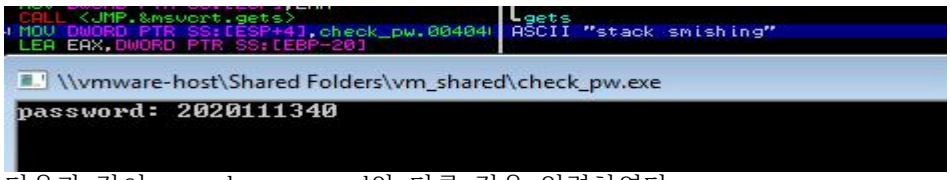
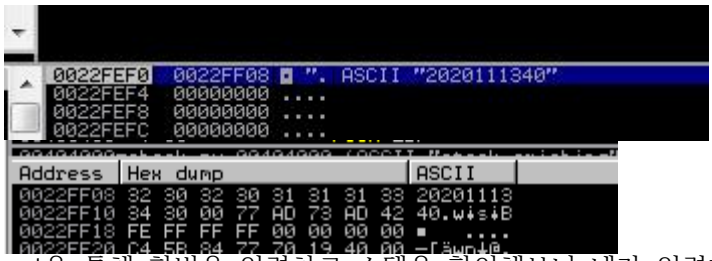
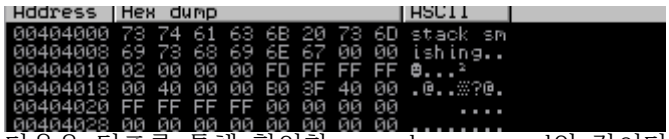
1 - 틀린 패스워드에 대한 실습



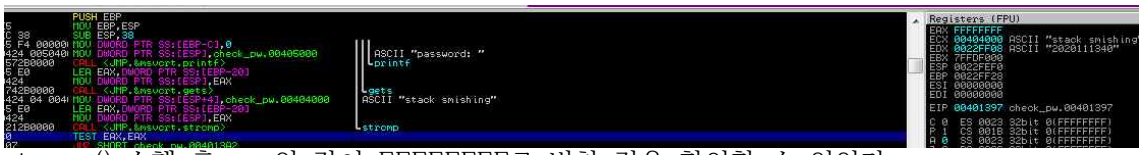
다음과 같이 saved_password와 다른 값을 입력하였다.



get을 통해 학번을 입력하고 스택을 확인해보니 내가 입력한 값이 올라가있는 것을 확인할 수 있다.

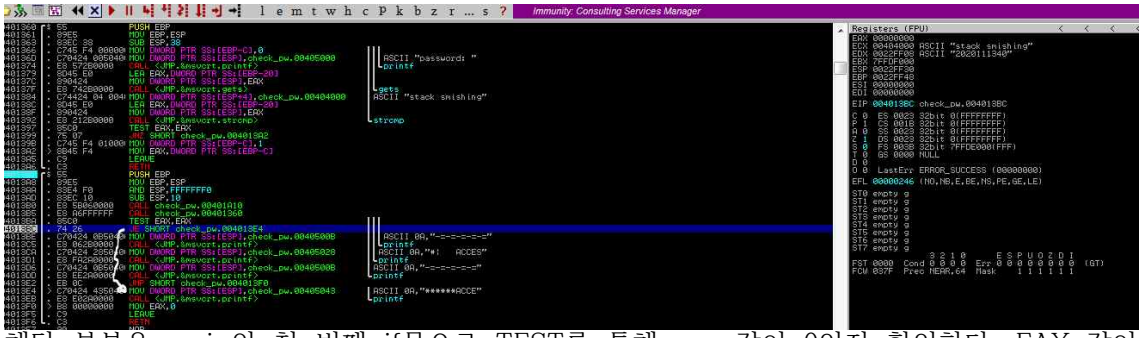


다음은 덤프를 통해 확인한 saved_password의 값이다.



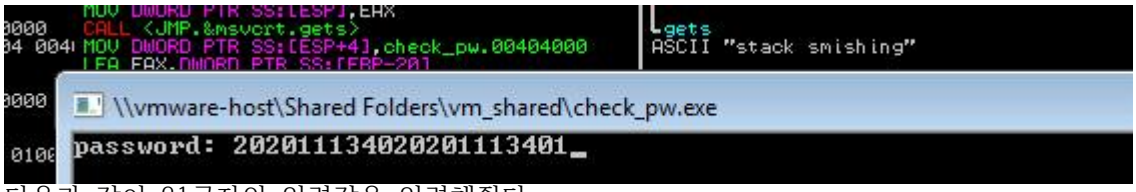
strcmp() 수행 후 eax의 값이 FFFFFFFF로 변한 것을 확인할 수 있었다.
그 후 TEST를 수행하여 eax 값이 0인지 확인하고 flag 값을 경우에 맞게 세팅한다.

해당 실습의 경우에는 틀린 패스워드를 입력했기 때문에 test eax, eax의 결과로 저장된 0 값이 0022FFC에 플래그 값 0이 저장된 것을 볼 수 있다.

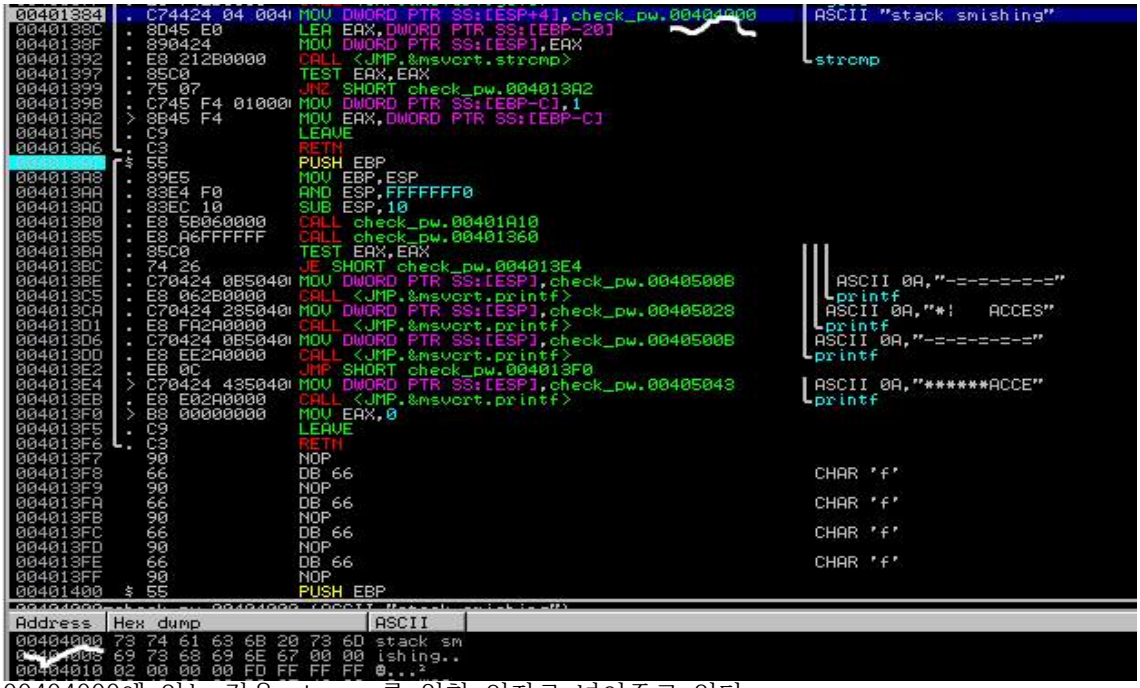


해당 부분은 main의 첫 번째 if문으로 TEST를 통해 eax 값이 0인지 확인한다. EAX 값이 0이라서 004013E4로 분기하여 ACCESS DENIED를 출력할 것이다.

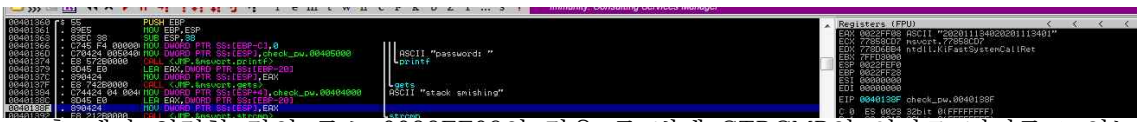
2 - 스택 오버플로우에 대한 실습



다음과 같이 21글자의 입력값을 입력해줬다.
버퍼의 크기는 20이라 마지막 1이 버퍼오버플로우가 일어나 다른 곳에 저장되게 될 것이다.

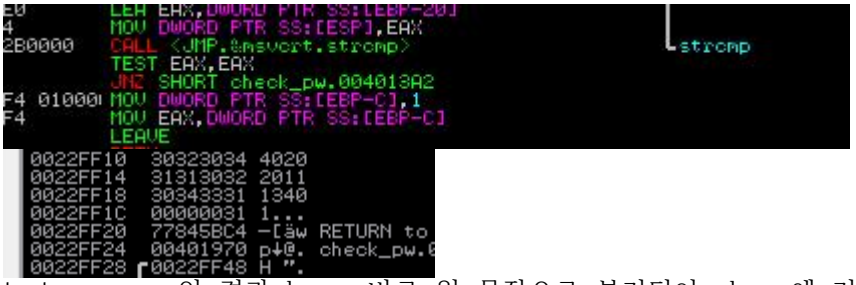


00404000에 있는 값을 strcmp를 위한 인자로 넣어주고 있다.
덤프에서 값이 확인 가능하다. 이 값은 saved_password이다.

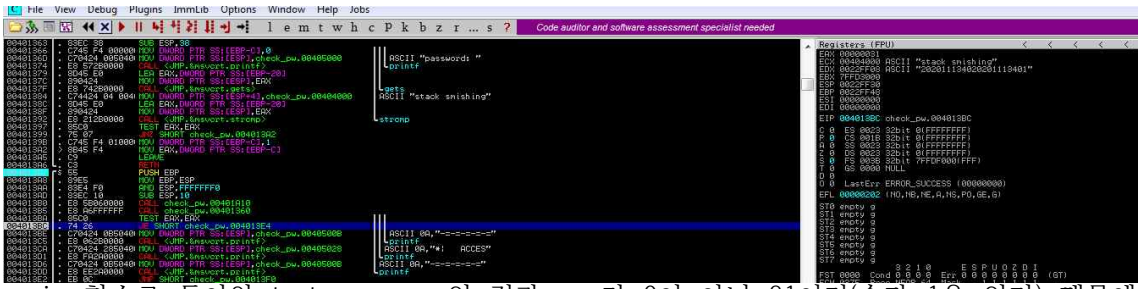


그 후 내가 입력한 값의 주소 0022FF08의 값을 두 번째 STRCMP의 인자로 넣어주고 있는 것을 확인할 수 있다.

그 후 strcmp의 결과로 eax에 FFFFFFFF가 저장된 것을 확인할 수 있다.



test eax, eax의 결과 leave 바로 윗 문장으로 분기되어 ebp-c에 기존 저장되어 있던 값이 eax에 저장된다. ebp-c에 기존 저장되어 있는 값은 버퍼 오버플로우가 일어나 1이 저장되어 있는 것을 확인할 수 있다.

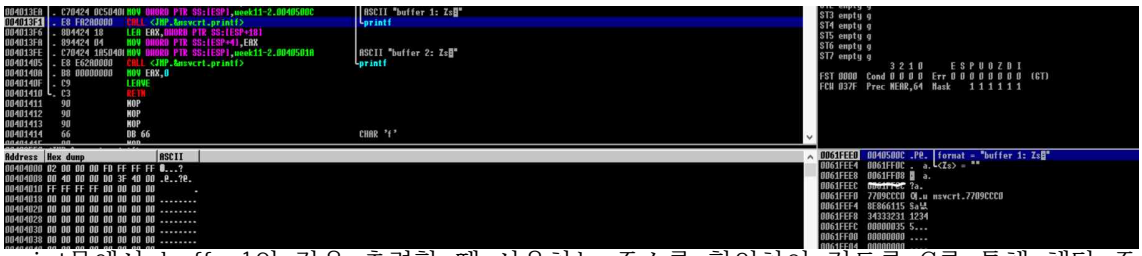


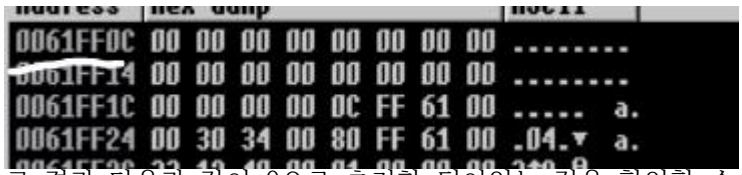
main 함수로 돌아와 test eax, eax의 결과 eax가 0이 아닌 31이기(숫자 1을 의미) 때문에 분기하지 않고 access access가 출력되고 종료되는 것을 확인할 수 있다.

3 - 강의록 마지막장

다음과 같이 exitprocess 바로 위에 있는 서브루틴을 F7을 이용해서 살펴보고 main 함수인 것을 확인했다.

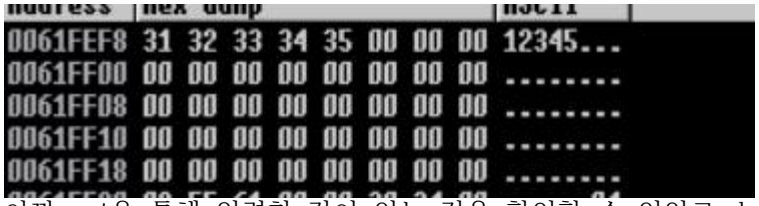
다음과 같이 get을 통해 12345를 입력하고


print문에서 buffer1의 값을 출력할 때 사용하는 주소를 확인하여 컨트롤+G를 통해 해당 주소의 값을 확인해봤다.

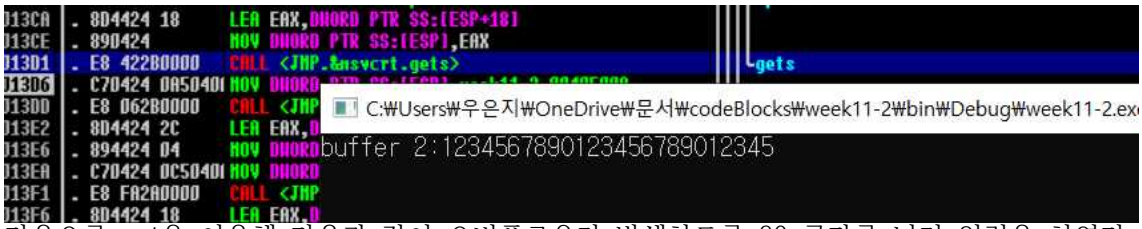


그 결과 다음과 같이 0으로 초기화 되어있는 것을 확인할 수 있다.
왜냐하면 코드에서 배열을 = {0,}으로 초기화 해줬기 때문이다.

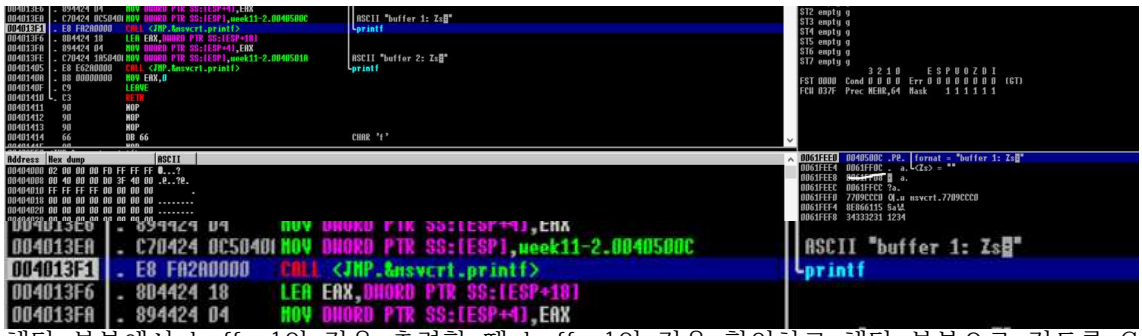

다음으론 buffer2를 printf 해주는 곳으로 가서 buffer2의 값을 출력할 때 참조하는 주소를 알아온후 덤프에서 확인해봤다.



아까 get을 통해 입력한 값이 있는 것을 확인할 수 있었고, buffer2 밑에 바로 buffer1이 저장된다는 사실도 알 수 있었다.



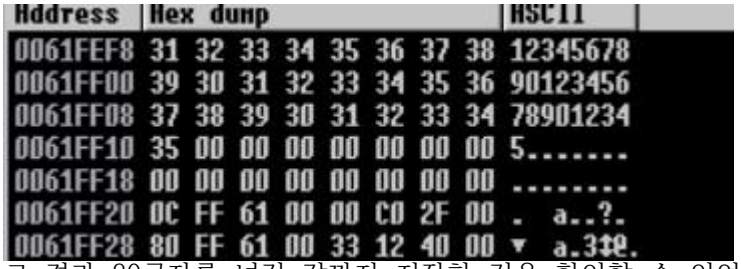
다음으론 get을 이용해 다음과 같이 오버플로우가 발생하도록 20 글자를 넘겨 입력을 하였다.



해당 부분에서 buffer1의 값을 출력할 때 buffer1의 값을 확인하고 해당 부분으로 컨트롤+G를 이용해 가서 값을 확인해봤다.

다음과 같이 12345라는 값이 들어와있다는 것을 확인할 수 있었다.
이 이유를 살펴보기 위해서 buffer2의 print 부분에 가서 덤프 값을 확인해보겠다.

다음과 같이 61FEF8에 buffer2의 값이 저장되어 있다는 것을 알 수 있었다.
해당 부분을 덤프로 가서 확인해보겠다.



그 결과 20글자를 넘긴 값까지 저장한 것을 확인할 수 있었고, 20글자에서 초과되는 부분은 buffer1의 저장공간에 저장되었다는 것을 확인할 수 있었다. 그렇기 때문에 오버플로우가 일어나 buffer1은 아무것도 저장하지 않았음에도 12345가 출력되는 것을 알 수 있었다.