

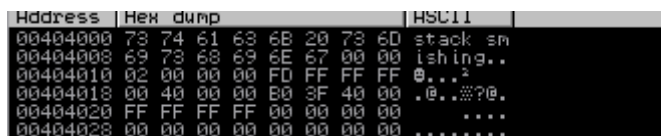
1 - 틀린 패스워드에 대한 실습



다음과 같이 saved_password와 다른 값을 입력하였다.



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

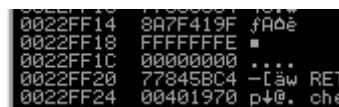


다음은 덤프를 통해 확인한 saved_password의 값이다.



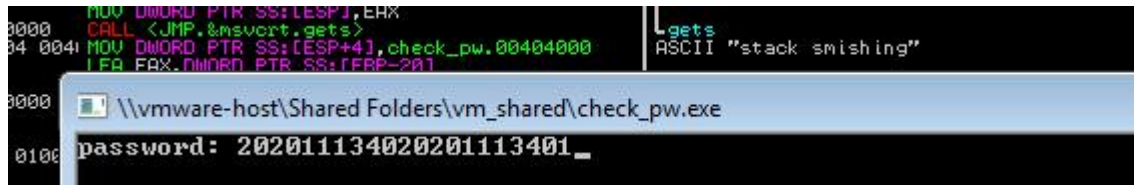
strcmp() 수행 후 eax의 값이 FFFFFFFF로 변한 것을 확인할 수 있었다.

그 후 TEST를 수행하여 eax 값이 0인지 확인하고 flag 값을 경우에 맞게 세팅한다.



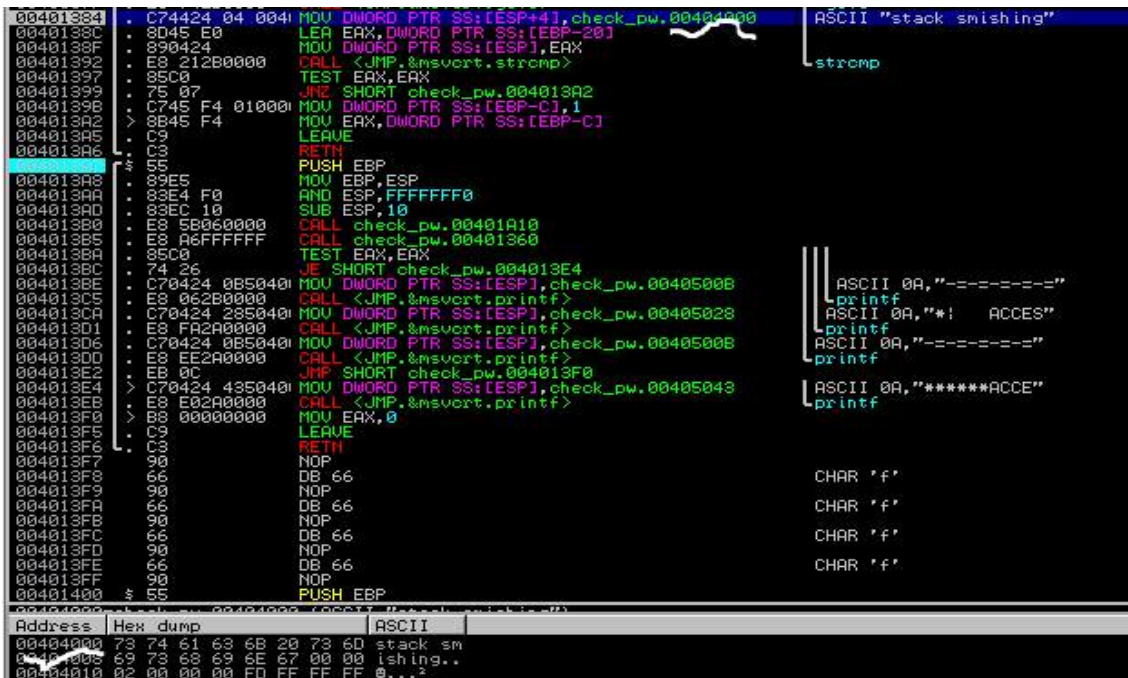
해당 실습의 경우에는 틀린 패스워드를 입력했기 때문에 test eax, eax의 결과로 저장된 0 값이 0022FFC에 플래그 값 0이 저장된 것을 볼 수 있다.

2 - 스택 오버플로우에 대한 실습



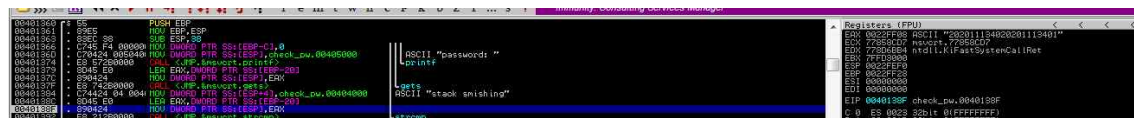
다음과 같이 21글자의 입력값을 입력해줬다.

버퍼의 크기는 20이라 마지막 1이 버퍼오버플로우가 일어나 다른 곳에 저장되게 될 것이다.

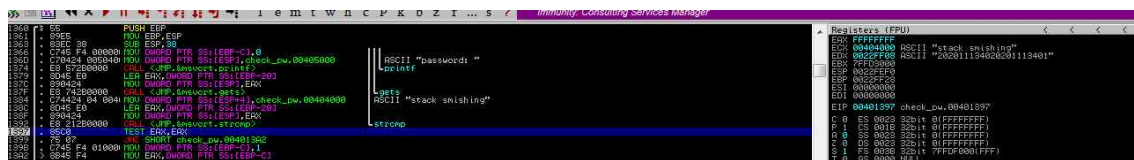


00404000에 있는 값을 strcmp를 위한 인자로 넣어주고 있다.

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.



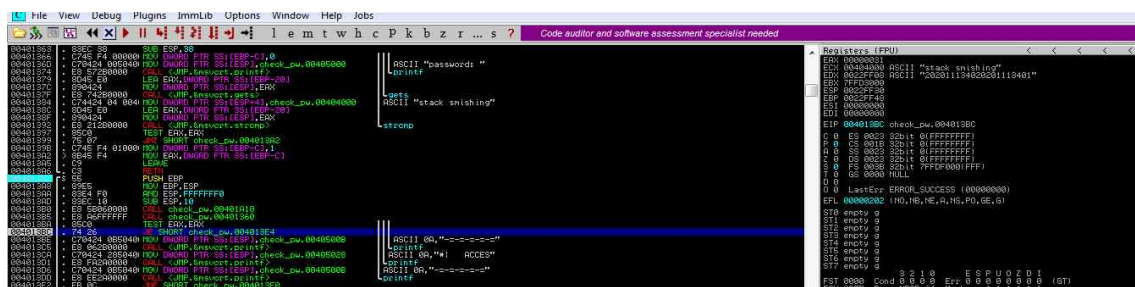
그 후 strcmp의 결과로 eax에 FFFFFFFF가 저장된 것을 확인할 수 있다.

```

E0      LEA EAX,DWORD PTR SS:[EBP-20]
4       MOV DWORD PTR SS:[ESP],EAX
2B0000  CALL <JMP.0x50000000>
        TEST EAX,EAX
        JNZ SHORT check_pw.004013A2
F4 010001 MOV DWORD PTR SS:[EBP-C],1
F4      MOV EAX,DWORD PTR SS:[EBP-C]
        LEAVE
0022FF10 30323034 4020
0022FF14 31313032 2011
0022FF18 30343331 1340
0022FF1C 00000031 1...
0022FF20 77845BC4 -[5w RETURN to
0022FF24 00401970 p+0. check_pw.0
0022FF28 0022FF48 H ".

```

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

```

00401222 . 894424 04      MOV DWORD PTR SS:[ESP+4],EAX
00401226 . A1 04704000     MOV EAX,DWORD PTR DS:[4070041]
00401228 . 890424         MOV DWORD PTR SS:[ESP],EAX
0040122E . E8 20010000     CALL week11-2.00401360
00401233 . 89C3          MOV EBX,EAX
00401235 . E8 26200000     CALL <JMP.&msvcrt._cexit>          |msvcrt._cexit
00401238 . 891C24         MOV DWORD PTR SS:[ESP],EBX
0040123D . E8 F20E0000     CALL <JMP.&KERNEL32.ExitProcess>   |ExitProcess
00401242 > 8B1D A4814000   MOV EBX,DWORD PTR DS:[<&msvcrt._iob>] |msvcrt._iob

```

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

```

LEA EAX,DWORD PTR SS:[ESP+10]
MOV DWORD PTR SS:[ESP],EAX
CALL <JMP.&msvcrt.gets>          |gets
MOV DWORD PTR SS:[ESP],week11-2.0040500A
CALL <JMP.&msvcrt.gets>

```

C:\Users\우은지\OneDrive\문서\codeBlocks\week11-2\bin\Debug\week11-2.exe

buffer 2:12345

다음과 같이 get을 통해 12345를 입력하고

```

004013E8 . C70424 0C5040  MOV DWORD PTR SS:[ESP],week11-2.0040500C |ASCII "buffer 1: %d"
004013EA . E8 F8200000     CALL <JMP.&msvcrt.printf>          |printf
004013F6 . 894424 04      MOV DWORD PTR SS:[ESP+4],EAX
004013FE . C70424 105040  MOV DWORD PTR SS:[ESP],week11-2.00405010 |ASCII "buffer 2: %d"
00401400 . 00 00000000     JZ <JMP.&msvcrt.printf>          |printf
00401408 . 00 00000000     JZ <JMP.&msvcrt.printf>
0040140F . C9            LEAVE
00401410 . C3            RETN
00401411 . 90            NOP
00401412 . 90            NOP
00401413 . 90            NOP
00401414 . 66            DB 66
00401415 . 00            DB 00

```

Address Hex dump ASCII

00401400	02 00 00 00 00 00 00 00
00401401	00 00 00 00 3F 40 00 00
00401402	FF FF FF FF 00 00 00 00
00401403	00 00 00 00 00 00 00 00
00401404	00 00 00 00 00 00 00 00
00401405	00 00 00 00 00 00 00 00
00401406	00 00 00 00 00 00 00 00
00401407	00 00 00 00 00 00 00 00
00401408	00 00 00 00 00 00 00 00
00401409	00 00 00 00 00 00 00 00
0040140A	00 00 00 00 00 00 00 00
0040140B	00 00 00 00 00 00 00 00
0040140C	00 00 00 00 00 00 00 00
0040140D	00 00 00 00 00 00 00 00
0040140E	00 00 00 00 00 00 00 00
0040140F	00 00 00 00 00 00 00 00
00401410	00 00 00 00 00 00 00 00
00401411	00 00 00 00 00 00 00 00
00401412	00 00 00 00 00 00 00 00
00401413	00 00 00 00 00 00 00 00
00401414	00 00 00 00 00 00 00 00
00401415	00 00 00 00 00 00 00 00
00401416	00 00 00 00 00 00 00 00
00401417	00 00 00 00 00 00 00 00
00401418	00 00 00 00 00 00 00 00
00401419	00 00 00 00 00 00 00 00
0040141A	00 00 00 00 00 00 00 00
0040141B	00 00 00 00 00 00 00 00
0040141C	00 00 00 00 00 00 00 00
0040141D	00 00 00 00 00 00 00 00
0040141E	00 00 00 00 00 00 00 00
0040141F	00 00 00 00 00 00 00 00
00401420	00 00 00 00 00 00 00 00
00401421	00 00 00 00 00 00 00 00
00401422	00 00 00 00 00 00 00 00
00401423	00 00 00 00 00 00 00 00
00401424	00 00 00 00 00 00 00 00
00401425	00 00 00 00 00 00 00 00
00401426	00 00 00 00 00 00 00 00
00401427	00 00 00 00 00 00 00 00
00401428	00 00 00 00 00 00 00 00
00401429	00 00 00 00 00 00 00 00
0040142A	00 00 00 00 00 00 00 00
0040142B	00 00 00 00 00 00 00 00
0040142C	00 00 00 00 00 00 00 00
0040142D	00 00 00 00 00 00 00 00
0040142E	00 00 00 00 00 00 00 00
0040142F	00 00 00 00 00 00 00 00
00401430	00 00 00 00 00 00 00 00
00401431	00 00 00 00 00 00 00 00
00401432	00 00 00 00 00 00 00 00
00401433	00 00 00 00 00 00 00 00
00401434	00 00 00 00 00 00 00 00
00401435	00 00 00 00 00 00 00 00
00401436	00 00 00 00 00 00 00 00
00401437	00 00 00 00 00 00 00 00
00401438	00 00 00 00 00 00 00 00
00401439	00 00 00 00 00 00 00 00
0040143A	00 00 00 00 00 00 00 00
0040143B	00 00 00 00 00 00 00 00
0040143C	00 00 00 00 00 00 00 00
0040143D	00 00 00 00 00 00 00 00
0040143E	00 00 00 00 00 00 00 00
0040143F	00 00 00 00 00 00 00 00
00401440	00 00 00 00 00 00 00 00
00401441	00 00 00 00 00 00 00 00
00401442	00 00 00 00 00 00 00 00
00401443	00 00 00 00 00 00 00 00
00401444	00 00 00 00 00 00 00 00
00401445	00 00 00 00 00 00 00 00
00401446	00 00 00 00 00 00 00 00
00401447	00 00 00 00 00 00 00 00
00401448	00 00 00 00 00 00 00 00
00401449	00 00 00 00 00 00 00 00
0040144A	00 00 00 00 00 00 00 00
0040144B	00 00 00 00 00 00 00 00
0040144C	00 00 00 00 00 00 00 00
0040144D	00 00 00 00 00 00 00 00
0040144E	00 00 00 00 00 00 00 00
0040144F	00 00 00 00 00 00 00 00
00401450	00 00 00 00 00 00 00 00
00401451	00 00 00 00 00 00 00 00
00401452	00 00 00 00 00 00 00 00
00401453	00 00 00 00 00 00 00 00
00401454	00 00 00 00 00 00 00 00
00401455	00 00 00 00 00 00 00 00
00401456	00 00 00 00 00 00 00 00
00401457	00 00 00 00 00 00 00 00
00401458	00 00 00 00 00 00 00 00
00401459	00 00 00 00 00 00 00 00
0040145A	00 00 00 00 00 00 00 00
0040145B	00 00 00 00 00 00 00 00
0040145C	00 00 00 00 00 00 00 00
0040145D	00 00 00 00 00 00 00 00
0040145E	00 00 00 00 00 00 00 00
0040145F	00 00 00 00 00 00 00 00
00401460	00 00 00 00 00 00 00 00
00401461	00 00 00 00 00 00 00 00
00401462	00 00 00 00 00 00 00 00
00401463	00 00 00 00 00 00 00 00
00401464	00 00 00 00 00 00 00 00
00401465	00 00 00 00 00 00 00 00
00401466	00 00 00 00 00 00 00 00
00401467	00 00 00 00 00 00 00 00
00401468	00 00 00 00 00 00 00 00
00401469	00 00 00 00 00 00 00 00
0040146A	00 00 00 00 00 00 00 00
0040146B	00 00 00 00 00 00 00 00
0040146C	00 00 00 00 00 00 00 00
0040146D	00 00 00 00 00 00 00 00
0040146E	00 00 00 00 00 00 00 00
0040146F	00 00 00 00 00 00 00 00
00401470	00 00 00 00 00 00 00 00
00401471	00 00 00 00 00 00 00 00
00401472	00 00 00 00 00 00 00 00
00401473	00 00 00 00 00 00 00 00
00401474	00 00 00 00 00 00 00 00
00401475	00 00 00 00 00 00 00 00
00401476	00 00 00 00 00 00 00 00
00401477	00 00 00 00 00 00 00 00
00401478	00 00 00 00 00 00 00 00
00401479	00 00 00 00 00 00 00 00
0040147A	00 00 00 00 00 00 00 00
0040147B	00 00 00 00 00 00 00 00
0040147C	00 00 00 00 00 00 00 00
0040147D	00 00 00 00 00 00 00 00
0040147E	00 00 00 00 00 00 00 00
0040147F	00 00 00 00 00 00 00 00
00401480	00 00 00 00 00 00 00 00
00401481	00 00 00 00 00 00 00 00
00401482	00 00 00 00 00 00 00 00
00401483	00 00 00 00 00 00 00 00
00401484	00 00 00 00 00 00 00 00
00401485	00 00 00 00 00 00 00 00
00401486	00 00 00 00 00 00 00 00
00401487	00 00 00 00 00 00 00 00
00401488	00 00 00 00 00 00 00 00
00401489	00 00 00 00 00 00 00 00
0040148A	00 00 00 00 00 00 00 00
0040148B	00 00 00 00 00 00 00 00
0040148C	00 00 00 00 00 00 00 00
0040148D	00 00 00 00 00 00 00 00
0040148E	00 00 00 00 00 00 00 00
0040148F	00 00 00 00 00 00 00 00
00401490	00 00 00 00 00 00 00 00
00401491	00 00 00 00 00 00 00 00
00401492	00 00 00 00 00 00 00 00
00401493	00 00 00 00 00 00 00 00
00401494	00 00 00 00 00 00 00 00
00401495	00 00 00 00 00 00 00 00
00401496	00 00 00 00 00 00 00 00
00401497	00 00 00 00 00 00 00 00
00401498	00 00 00 00 00 00 00 00
00401499	00 00 00 00 00 00 00 00
0040149A	00 00 00 00 00 00 00 00
0040149B	00 00 00 00 00 00 00 00
0040149C	00 00 00 00 00 00 00 00
0040149D	00 00 00 00 00 00 00 00
0040149E	00 00 00 00 00 00 00 00
0040149F	00 00 00 00 00 00 00 00
004014A0	00 00 00 00 00 00 00 00
004014A1	00 00 00 00 00 00 00 00
004014A2	00 00 00 00 00 00 00 00
004014A3	00 00 00 00 00 00 00 00
004014A4	00 00 00 00 00 00 00 00
004014A5	00 00 00 00 00 00 00 00
004014A6	00 00 00 00 00 00 00 00
004014A7	00 00 00 00 00 00 00 00
004014A8	00 00 00 00 00 00 00 00
004014A9	00 00 00 00 00 00 00 00
004014AA	00 00 00 00 00 00 00 00
004014AB	00 00 00 00 00 00 00 00
004014AC	00 00 00 00 00 00 00 00
004014AD	00 00 00 00 00 00 00 00
004014AE	00 00 00 00 00 00 00 00
004014AF	00 00 00 00 00 00 00 00
004014B0	00 00 00 00 00 00 00 00
004014B1	00 00 00 00 00 00 00 00
004014B2	00 00 00 00 00 00 00 00
004014B3	00 00 00 00 00 00 00 00
004014B4	00 00 00 00 00 00 00 00
004014B5	00 00 00 00 00 00 00 00
004014B6	00 00 00 00 00 00 00 00
004014B7	00 00 00 00 00 00 00 00
004014B8	00 00 00 00 00 00 00 00
004014B9	00 00 00 00 00 00 00 00
004014BA	00 00 00 00 00 00 00 00
004014BB	00 00 00 00 00 00 00 00
004014BC	00 00 00 00 00 00 00 00
004014BD	00 00 00 00 00 00 00 00
004014BE	00 00 00 00 00 00 00 00
004014BF	00 00 00 00 00 00 00 00
004014C0	00 00 00 00 00 00 00 00
004014C1	00 00 00 00 00 00 00 00
004014C2	00 00 00 00 00 00 00 00
004014C3	00 00 00 00 00 00 00 00
004014C4	00 00 00 00 00 00 00 00
004014C5	00 00 00 00 00 00 00 00
004014C6	00 00 00 00 00 00 00 00
004014C7	00 00 00 00 00 00 00 00
004014C8	00 00 00 00 00 00 00 00
004014C9	00 00 00 00 00 00 00 00
004014CA	00 00 00 00 00 00 00 00
004014CB	00 00 00 00 00 00 00 00
004014CC	00 00 00 00 00 00 00 00
004014CD	00 00 00 00 00 00 00 00
004014CE	00 00 00 00 00 00 00 00
004014CF	00 00 00 00 00 00 00 00
004014D0	00 00 00 00 00 00 00 00
004014D1	00 00 00 00 00 00 00 00
004014D2	00 00 00 00 00 00 00 00
004014D3	00 00 00 00 00 00 00 00
004014D4	00 00 00 00 00 00 00 00
004014D5	00 00 00 00 00 00 00 00
004014D6	00 00 00 00 00 00 00 00
004014D7	00 00 00 00 00 00 00 00
004014D8	00 00 00 00 00 00 00 00

Address	Hex	Dump	ASCII
0061FEF8	31 32 33 34 35 00 00 00	12345...	
0061FF00	00 00 00 00 00 00 00 00	
0061FF08	00 00 00 00 00 00 00 00	
0061FF10	00 00 00 00 00 00 00 00	
0061FF18	00 00 00 00 00 00 00 00	

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

```

013C8 . 804424 18    LEA EAX, DWORD PTR SS:[ESP+18]
013CE . 890424      MOV DWORD PTR SS:[ESP], EAX
013D1 . E8 422B0000 CALL <JMP.&msvcrt.gets>
013D6 . C70424 0A5040 MOV DWORD PTR SS:[ESP], 0A5040
013D8 . E8 062B0000 CALL <JMP.>
013E2 . 804424 2C    LEA EAX, 2C
013E6 . 894424 04    MOV DWORD PTR SS:[ESP+4], EAX
013EA . C70424 0C5040 MOV DWORD PTR SS:[ESP], 0C5040
013F1 . E8 FA2A0000 CALL <JMP.>
013F6 . 804424 18    LEA EAX, 18

```

buffer 2: 1234567890123456789012345

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

```

004013E0 . 894424 04    MOV DWORD PTR SS:[ESP+4], EAX
004013E1 . C70424 0C5040 MOV DWORD PTR SS:[ESP], 0C5040
004013F1 . E8 FA2A0000 CALL <JMP.&msvcrt.printf>
004013F6 . 804424 18    LEA EAX, DWORD PTR SS:[ESP+18]
004013FA . 894424 04    MOV DWORD PTR SS:[ESP+4], EAX

```

ASCII "buffer 1: Zs"

ASCII "buffer 2: Zs"

printf

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

```

004013E0 . 894424 04    MOV DWORD PTR SS:[ESP+4], EAX
004013E1 . C70424 0C5040 MOV DWORD PTR SS:[ESP], 0C5040
004013F1 . E8 FA2A0000 CALL <JMP.&msvcrt.printf>
004013F6 . 804424 18    LEA EAX, DWORD PTR SS:[ESP+18]
004013FA . 894424 04    MOV DWORD PTR SS:[ESP+4], EAX

```

ASCII "buffer 1: Zs"

printf

다음과 같이 12345라는 값이 들어와있다는 것을 확인할 수 있었다.

이 이유를 살펴보기 위해서 buffer2의 print 부분에 가서 덤프 값을 확인해보겠다.

```

004013E0 . 894424 04    MOV DWORD PTR SS:[ESP+4], EAX
004013E1 . C70424 0C5040 MOV DWORD PTR SS:[ESP], 0C5040
004013F1 . E8 FA2A0000 CALL <JMP.&msvcrt.printf>
004013F6 . 804424 18    LEA EAX, DWORD PTR SS:[ESP+18]
004013FA . 894424 04    MOV DWORD PTR SS:[ESP+4], EAX

```

ASCII "buffer 2: Zs"

printf

다음과 같이 61FEF8에 buffer2의 값이 저장되어 있다는 것을 알 수 있었다.

해당 부분을 덤프로 가서 확인해보겠다.

Address	Hex dump	ASCII
0061FEF8	31 32 33 34 35 36 37 38	12345678
0061FF00	39 30 31 32 33 34 35 36	90123456
0061FF08	37 38 39 30 31 32 33 34	78901234
0061FF10	35 00 00 00 00 00 00 00	5.....
0061FF18	00 00 00 00 00 00 00 00
0061FF20	0C FF 61 00 00 C0 2F 00	. a..?.
0061FF28	80 FF 61 00 33 12 40 00	a.3t.

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.

```

<JMP.&msvcrt.gets>
DWORD PTR [EIP+00000001] = 00405000
C:\Users\우은지\OneDrive\문서\codeBlocks\week11-2\bin\W
buffer 2:1234567890123456789012345
buffer 1: 12345
buffer 2: 1234567890123456789012345

```