Program analysis

Tuesday, 24 March 2020 3:23 PM

First input
Rbp - 0x1 : A(0x41)

```
gef➤  x/4gx $rbp -16
0x7fffffffe410: 0x00007fffffffe500      0x4100000000000000
0x7fffffffe420: 0x0000000000400710      0x00007ffff7a05b97
gef➤  x/gx $rbp
0x7fffffffe420: 0x0000000000400710
```

2nd input
Rbp - 0x2:

```
 →   0x400684 <main+93>        lea    rax, [rbp-0x2]
     0x400688 <main+97>        mov    rsi, rax
     0x40068b <main+100>       lea    rdi, [rip+0x13d]        # 0x4007cf
     0x400692 <main+107>       mov    eax, 0x0
     0x400697 <main+112>       call   0x400530 <__isoc99_scanf@plt>
```

1st and 2nd input in memory

```
gef➤  x/4gx $rbp-16
0x7fffffffe410: 0x00007fffffffe500      0x4142000000000000
0x7fffffffe420: 0x0000000000400710      0x00007ffff7a05b97
gef➤
```

Jump if less or equal

```
 →   0x40069c <main+117>       movzx  edx, BYTE PTR [rbp-0x1]
     0x4006a0 <main+121>       movzx  eax, BYTE PTR [rbp-0x2]
     0x4006a4 <main+125>       cmp    dl, al
     0x4006a6 <main+127>       jle    0x4006cb <main+164>
```

0x41 is lesser than 0x42, so jump is taken

```
gef➤  i r $edx
edx            0x41     0x41
gef➤  i r $eax
eax            0x42     0x42
gef➤
```

```
$eflags: [zero CARRY PARITY ADJUST SIGN trap INTERRUPT direction overflow resume virtualx86 identification]
```

0x42 is not lesser than 0x41, so jump is not taken

```
  0x4006cb <main+164>       movzx  edx, BYTE PTR [rbp-0x2]
  0x4006cf <main+168>       movzx  eax, BYTE PTR [rbp-0x1]
  0x4006d3 <main+172>       cmp    dl, al
  0x4006d5 <main+174>       jle    0x4006fa <main+211>
```

```
gef➤  i r $edx
edx            0x42     0x42
gef➤  i r $eax
eax            0x41     0x41
gef➤
```