Catchme

周子涵

```
__int64 sub_D78()
{
  unsigned int n4; // [rsp+Ch] [rbp-14h]
  char nptr[8]; // [rsp+10h] [rbp-10h] BYREF
  unsigned __int64 v3; // [rsp+18h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  puts("index:");
  sub_A81(nptr);
  n4 = atoi(nptr);
  if ( n4 <= 4 && qword_202060[n4] )
  {
    free((void *)qword_202060[n4]);
    return 0;
  }
  else
  {
    puts("invalid operation");
    return 0xFFFFFFFFLL;
  }
}
```

发现 uaf 漏洞
应执行地址清零操作
尝试 nop free 函数失败和 mov qword ptr [rax+rdx], 0 清空指针失败
找可执行段

```
pwndbg> vmmap
LEGEND: STACK | HEAP | CODE | DATA | WX | RODATA
             Start                End Perm     Size  Offset File (set vmmap-prefer-relpaths on)
    0x555555400000     0x555555402000 r-xp     2000       0 catchme
    0x555555601000     0x555555602000 r--p     1000    1000 catchme
    0x555555602000     0x555555603000 rw-p     1000    2000 catchme
    0x555555603000     0x555555624000 rw-p    21000       0 [heap]
    0x7ffff7d82000     0x7ffff7d85000 rw-p     3000       0 [anon_7ffff7d82]
    0x7ffff7d85000     0x7ffff7dad000 r--p    28000       0 /usr/lib/x86_64-linux-gnu/libc.so.6
    0x7ffff7dad000     0x7ffff7f42000 r-xp   195000   28000 /usr/lib/x86_64-linux-gnu/libc.so.6
    0x7ffff7f42000     0x7ffff7f9a000 r--p    58000  1bd000 /usr/lib/x86_64-linux-gnu/libc.so.6
    0x7ffff7f9a000     0x7ffff7f9b000 ---p     1000  215000 /usr/lib/x86_64-linux-gnu/libc.so.6
    0x7ffff7f9b000     0x7ffff7f9f000 r--p     4000  215000 /usr/lib/x86_64-linux-gnu/libc.so.6
    0x7ffff7f9f000     0x7ffff7fa1000 rw-p     2000  219000 /usr/lib/x86_64-linux-gnu/libc.so.6
    0x7ffff7fa1000     0x7ffff7fae000 rw-p     d000       0 [anon_7ffff7fa1]
    0x7ffff7fbb000     0x7ffff7fbd000 rw-p     2000       0 [anon_7ffff7fbb]
    0x7ffff7fbd000     0x7ffff7fc1000 r--p     4000       0 [vvar]
    0x7ffff7fc1000     0x7ffff7fc3000 r-xp     2000       0 [vdso]
    0x7ffff7fc3000     0x7ffff7fc5000 r--p     2000       0 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
    0x7ffff7fc5000     0x7ffff7fef000 r-xp    2a000    2000 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
    0x7ffff7fef000     0x7ffff7ffa000 r--p     b000   2c000 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
    0x7ffff7ffb000     0x7ffff7ffd000 r--p     2000   37000 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
    0x7ffff7ffd000     0x7ffff7fff000 rw-p     2000   39000 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
    0x7ffffffdd000     0x7fffffff000 rw-p    22000       0 [stack]
pwndbg>
```

找到 eh_frame 上数据使用较少的位置进行跳转

```
00000000014D3                   db 0FFh
00000000014D4                   db  2Bh ; +
00000000014D5 addr2             db    0                 ; CODE XREF: sub_D78+A1↑j
00000000014D6                   db    0
00000000014D7                   db    0
00000000014D8                   db    0
00000000014D9                   db    0
00000000014DA                   db    0
00000000014DB                   db    0
00000000014DC                   db    0
00000000014DD                   db    0
00000000014DE                   db    0
00000000014DF                   db    0
00000000014E0                   db  14h
00000000014E1                   db    0
00000000014E2                   db    0
00000000014E3                   db    0
00000000014E4                   db    0
00000000014E5                   db    0
00000000014E6                   db    0
00000000014E7                   db    0
00000000014E8                   db    1
00000000014E9                   db  7Ah ; z
00000000014EA                   db  52h ; R
00000000014EB                   db    0
00000000014EC                   db    1
00000000014ED                   db  78h ; x
00000000014EE                   db  10h
00000000014EF                   db    1
00000000014F0                   db  1Bh
```

改为

```
00000000014D2                   db 0FFh
00000000014D3                   db 0FFh
00000000014D4                   db  2Bh ; +
00000000014D5 addr2             db  48h ; H             ; CODE XREF: sub_D78+A1↑j
00000000014D5                                           ; Keypatch modified this from:
00000000014D5                                           ;   db 0
00000000014D5                                           ;   db 0
00000000014D5                                           ;   db 0
00000000014D5                                           ;   db 0
00000000014D6                   db  8Bh
00000000014D7                   db  0Ch
00000000014D8                   db    2
00000000014D9                   db  48h ; H             ; Keypatch modified this from:
00000000014D9                                           ;   db 0
00000000014D9                                           ;   db 0
00000000014D9                                           ;   db 0
00000000014DA                   db  89h
00000000014DB                   db 0CFh
00000000014DC                   db  48h ; H             ; Keypatch modified this from:
00000000014DC                                           ;   db 0
00000000014DC                                           ;   db 0
00000000014DC                                           ;   db 0
00000000014DC                                           ;   db 0
00000000014DC                                           ;   db 14h
00000000014DC                                           ;   db 0
00000000014DC                                           ;   db 0
00000000014DC                                           ;   db 0
00000000014DD                   db 0C7h
00000000014DE                   db    4
00000000014DF                   db  10h
00000000014E0                   db    0
00000000014E1                   db    0
00000000014E2                   db    0
00000000014E3                   db    0
00000000014E4                   db 0E9h                 ; Keypatch modified this from:
00000000014E4                                           ;   db 0
00000000014E4                                           ;   db 0
00000000014E4                                           ;   db 0
00000000014E4                                           ;   db 0
00000000014E4                                           ;   db 1
00000000014E5                   db  37h ; 7
00000000014E6                   db 0F9h
00000000014E7                   db 0FFh
00000000014E8                   db 0FFh
```

mov rcx,[rdx+rax]

mov rdi,rcx

mov qword ptr [rax+rdx], 0

jmp addr1

执行完清零操作之后再跳转回去 free

```
0000000000000E05                mov     eax, [rbp+var_14]
0000000000000E08                cdqe
0000000000000E0A                lea     rdx, ds:0[rax*8]
0000000000000E12                lea     rax, qword_202060
0000000000000E19                jmp     near ptr addr2  ; Keypatch modified this from:
0000000000000E19                                        ;   mov rax, [rdx+rax]
0000000000000E19                                        ;   mov rdi, rax
0000000000000E19                                        ; Keypatch padded NOP to next boundary: 2 bytes
0000000000000E19 ; ---------------------------------------------------------------------------
0000000000000E1E                db  90h
0000000000000E1F                db  90h
0000000000000E20 ; ---------------------------------------------------------------------------
0000000000000E20
0000000000000E20 addr1:
0000000000000E20                call    _free
```