

## mvm 防御

整数溢出，将检测 $(\text{__QWORD}j + 1) > 6$ 的jle patch为jbe即可。

```
.text:0000000000001502          loc_1502:                                ; CODE XREF: main+2341j
.text:0000000000001502                                ; DATA XREF: .rodata:jpt_145D+0
.text:0000000000001502 48 83 BD 90 FF FD FF 00    cmp     [rbp+vm_sp], 0          ; jumtable 000000000000145D case 3
.text:000000000000150A 75 2A                                jnz     short loc_1536
.text:000000000000150A                                ;
.text:000000000000150C 48 8B 05 2D 3B 00 00    mov     rax, cs:stderr
.text:0000000000001513 48 89 C1                                mov     rcx, rax                ; s
.text:0000000000001516 BA 12 00 00 00    mov     edx, 12h               ; n
.text:000000000000151B BE 01 00 00 00    mov     esi, 1                 ; size
.text:0000000000001520 48 8D 3D 2B 1B 00 00    lea     rdi, aStackUnderflow   ; "Stack Underflow !\n"
.text:0000000000001527 E8 04 FC FF FF          call    _fwrite
.text:0000000000001527                                ;
.text:000000000000152C BF 00 00 00 00    mov     edi, 0                 ; status
.text:0000000000001531 E8 8A FB FF FF          call    __exit
.text:0000000000001531                                ;
.text:0000000000001536                                ; -----
.text:0000000000001536                                ;
.text:0000000000001536          loc_1536:                                ; CODE XREF: main+2E11j
.text:0000000000001536 48 8B 85 98 FF FD FF    mov     rax, [rbp+var_20068]
.text:000000000000153D 48 8B 40 08            mov     rax, [rax+8]
.text:0000000000001541 48 83 F8 06            cmp     rax, 6
.text:0000000000001545 76 2A                                jbe     short loc_1571          ; Keypatch modified this from:
.text:0000000000001545                                ;     jle short loc_1571
.text:0000000000001545
```

## diff 防御

read函数会造成栈溢出，将read的参数从128patch为123即可。

```
unsigned __int64 __fastcall sub_1566(__int64 a1, __int64 a2)
{
    int v3; // [rsp+1Ch] [rbp-14h] BYREF
    void *buf; // [rsp+20h] [rbp-10h]
    unsigned __int64 v5; // [rsp+28h] [rbp-8h]

    v5 = __readfsqword(0x28u);
    v3 = 1;
    printf("file 1 or 2:");
    __isoc99_scanf("%d", &v3);
    if ( v3 == 1 )
    {
        buf = (void *)(a1 + 5);
    }
    else
    {
        if ( v3 != 2 )
            return __readfsqword(0x28u) ^ v5;
        buf = (void *)(a2 + 5);
    }
    printf("filename:");
    read(0, buf, 123uLL);
}
```

## szp2 防御

用 <https://github.com/TTY-flag/evilPatcher> 上的工具加通防即可。

```
python2 evilPatcher/evilPatcher.py ./szp2 evilPatcher/sandboxes/mini_sandbox.asm
```