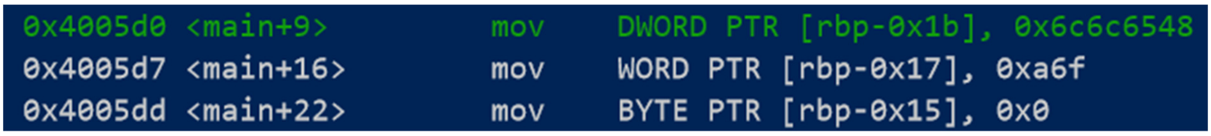
Disassembly of program

Store “Hello\n” on the stack

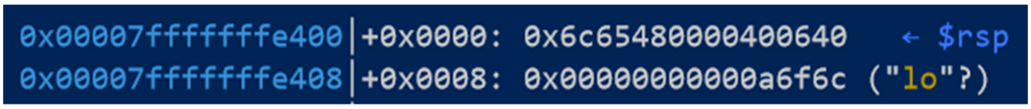
1st storage -> “Hell” (dword since it is 4 bytes)

2nd storage -> “o\n”(word since it is 2 bytes)

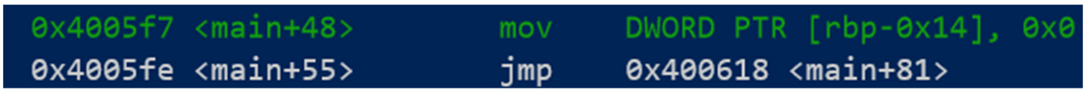
3rd storage -> null terminator(1 byte 0x0)



```
0x4005d0 <main+9>          mov    DWORD PTR [rbp-0x1b], 0x6c6c6548
0x4005d7 <main+16>         mov    WORD PTR [rbp-0x17], 0xa6f
0x4005dd <main+22>         mov    BYTE PTR [rbp-0x15], 0x0
```



```
0x00007fffffffe400│+0x0000: 0x6c65480000400640   ← $rsp
0x00007fffffffe408│+0x0008: 0x0000000000a6f6c ("lo"?)
```

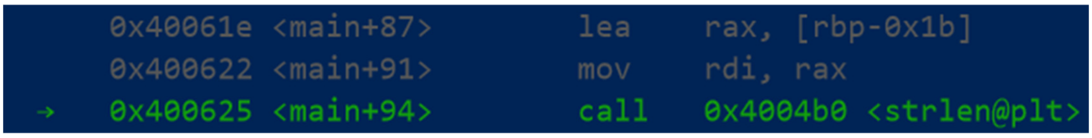


```
0x4005f7 <main+48>         mov    DWORD PTR [rbp-0x14], 0x0
0x4005fe <main+55>         jmp    0x400618 <main+81>
```

1st instruction -> store argument “Hello\n” in rax register

2nd instruction -> prepare for function call as rdi is populated with “Hello\n”

3rd instruction -> calls strlen()



```
     0x40061e <main+87>          lea    rax, [rbp-0x1b]
     0x400622 <main+91>          mov    rdi, rax
 →   0x400625 <main+94>          call   0x4004b0 <strlen@plt>
```