Debugging walkthrough: Access violation on nonsense instruction, episode 3

devblogs.microsoft.com/oldnewthing/20150828-00

August 28, 2015

Raymond Chen

A colleague of mine asked for help debugging a strange failure. Execution halted on what appeared to be a nonsense instruction.

```
eax=022b13a0 ebx=00000000 ecx=02570df4 edx=769f4544 esi=02570dec edi=05579748
eip=76c49131 esp=05cce038 ebp=05cce07c iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010202
KERNELBASE!GetFileAttributesExW+0x2:
76c49131 ec              in      al,dx
```

This is clearly an invalid instruction. But observe that the offset is +2, which is normally the start of the function, because the first two bytes of Windows operating system functions are a `mov edi, edi` instruction. Therefore, the function is corrupted. Lets look back two bytes to see if it gives any clues.

```
0:006> u 76c49131-2
KERNELBASE!GetFileAttributesExW:
76c4912f e95aecebf3      jmp     IoLog!Mine_GetFileAttributesExW (6ab07d8e)
```

Oh look, somebody is doing API patching (already unsupported) and they did a bad job. They tried to patch code while a thread was in the middle of executing it, resulting in a garbage instruction.

This is a bug in IoLog. The great thing about API patching is that when you screw up, it looks like an OS bug. That way, nobody ever files bugs against you!

(In this case, IoLog is a diagnostic tool which is logging file I/O performed by an application which is being instrumented.)

My colleague replied, “Thanks. Looks like a missing lock in IoLog. It doesn’t surprise me that API patching isn’t supported…”

Raymond Chen

Follow