Finding the constructor by scanning memory for the vtable

devblogs.microsoft.com/oldnewthing/20150320-00

March 20, 2015

Raymond Chen

In Looking for leaked objects by their vtable, we used the object's constructor to locate the vtable, and then scanned the heap for the vtable to find the leaked object. But you can run this technique in reverse, too.

Suppose you found an object and you want to find its constructor. This is not a problem if you have the source code, but if you are doing some reverse-engineering for application compatibility purposes, you don't have the luxury of the application source code. You may have figured out that the application fails because the byte at offset 0x50 is zero, but on the previous version of Windows, it was nonzero. You want to find out who sets the byte at offset 0x50, so that you can see why it is setting it to zero instead of a nonzero value.

If the object has a vtable, you can scan the code segments for a copy of the vtable. It will show up in an instruction like

```
mov dword ptr [reg], vtable_address
```

This is almost certainly the object's constructor, setting up the object vtable as part of construction. You can set a breakpoint here to break when the object is constructed, and then you can set a write breakpoint on offset 0x50 to see where its value is seto.

Raymond Chen

Follow