Why double-null-terminated strings instead of an array of pointers to strings?

devblogs.microsoft.com/oldnewthing/20110511-00

May 11, 2011

Raymond Chen

I mentioned this in passing in my description of the format of double-null-terminated strings, but I think it deserves calling out. Double-null-terminated strings may be difficult to create and modify, but they are very easy to serialize: You just write out the bytes as a blob. This property is very convenient when you have to copy around the list of strings: Transferring the strings is a simple matter of transferring the memory block as-is. No conversion is necessary. This makes it easy to do things like wrap the memory inside another container that supports only flat blobs of memory. As it turns out, a flat blob of memory is convenient in many ways. You can copy it around with memcpy. (This is important when capturing values across security boundaries.) You can save it to a file or into the registry as-is. It marshals very easily. It becomes possible to store it in an `IDataObject`. It can be freed with a single call. And in the cases where you can't allocate any memory at all (*e.g.*, you're filling a buffer provided by the caller), it's one of the few options available. This is also why self-relative security descriptors are so popular in Windows: Unlike absolute security descriptors, self-relative security descriptors can be passed around as binary blobs, which makes them easy to marshal, especially if you need to pass one from kernel mode to user mode. A single memory block with an array of integers containing offsets would also work, but as the commenter noted, it's even more cumbersome than double-null-terminated strings.

Mind you, if you don't need to marshal the list of strings (because it never crosses a security boundary and never needs to be serialized), then an array of string pointers works just fine. If you look around Win32, you'll find that most cases where double-null terminated strings exist are for the most part either inherited from 16-bit Windows or are one of the cases where marshalling is necessary.

Raymond Chen

Follow