How you might be loading a DLL during DLL_PROCESS_DETACH without even realizing it

devblogs.microsoft.com/oldnewthing/20100115-00

January 15, 2010

Raymond Chen

As you are I'm sure aware, you shouldn't be doing much of anything in your `DllMain` function, but you have to watch out for cases where you end up doing them accidentally.

Some time ago, I was investigating a failure which was traced back to loading a DLL inside `DLL_PROCESS_DETACH`. Wait, what kind of insane person *loads* a DLL as part of shutting down? Shouldn't you be cleaning up stuff, not creating new stuff?

The following is not the actual code, but it captures the same spirit:

```
INFO *CachedInfo;
BOOL WINAPI DllMain(HINSTANCE hinst, DWORD dwReason, void *pvReserved)
{
  switch (dwReason) {
  ...
  case DLL_PROCESS_DETACH:
    ...
    CoTaskMemFree(CachedInfo);
    ...
  }
  return TRUE;
}
```

There is some global variable that contains a pointer to memory that was allocated by `CoTaskMemAlloc`. In this case, I made it a cache, but the details aren't important. When the DLL is detached from the process, we free the cached memory so we don't have a leak. Since it is okay to pass `NULL` to the `CoTaskMemFree` function (it simply returns without doing anything), the cleanup code works even if we never called a function that put a value into the cache.

Except that this code ended up loading a DLL. The reason is delay-loading.

The authors of this DLL sped up its load time by marking `OLEAUT32.DLL` as a delay-loaded DLL, which means that it doesn't get loaded until somebody calls a function in it.

And in fact, nobody called a function from `OLEAUT32`. Ever.

"Hooray!" you shout. "We avoided loading `OLEAUT32` altogether." After all, the fastest code is code that doesn't run.

Except that it does run. Right there. In your `DLL_PROCESS_DETACH` handler.

Since nobody called a function from `OLEAUT32`, the call to `CoTaskMemFree` was the first call to `OLEAUT32` and therefore *caused it to be loaded*. From inside a `DLL_PROCESS_DETACH` handler.

Delay-loading is one of those features that is very convenient and saves you a lot of typing (namely, writing those stub functions yourself), but you also have to understand what's going on so you don't use it incorrectly. (In this case, a superficially-redundant `if (CachedInfo != NULL)` test needs to be inserted.)

Raymond Chen

Follow