Have you found any TheDailyWTF-worthy code during the development of Windows 95?

devblogs.microsoft.com/oldnewthing/20121203-00

December 3, 2012

Raymond Chen

Mott555 is interested in some sloppy/ugly code or strange workarounds or code comments during the development of Windows 95, like “anything TheDailyWTF-worthy.”

I discovered that opening a particular program churned the hard drive a lot when you opened it. I decided to hook up the debugger to see what the problem was. What I discovered was code that went roughly like this, in pseudo-code:

```
int TryToCallFunctionX(a, b, c)
{
  for each file in (SystemDirectory,
                    WindowsDirectory,
                    ProgramFilesDirectory(RecursiveSearch),
                    KitchenSink,
                    Uncle.GetKitchenSink)
  {
    hInstance = LoadLibrary(file);
    fn = GetProcAddress(hInstance, "FunctionX");
    if (fn != nullptr) {
        int result = fn(a,b,c);
        FreeLibrary(hInstance);
        return result;
    }
    fn = GetProcAddress(hInstance, "__imp__FunctionX");
    if (fn != nullptr) {
        int result = fn(a,b,c);
        FreeLibrary(hInstance);
        return result;
    }
    fn = GetProcAddress(hInstance, "FunctionX@12");
    if (fn != nullptr) {
        int result = fn(a,b,c);
        FreeLibrary(hInstance);
        return result;
    }
    fn = GetProcAddress(hInstance, "__imp__FunctionX@12");
    if (fn != nullptr) {
        int result = fn(a,b,c);
        FreeLibrary(hInstance);
        return result;
    }
    FreeLibrary(hInstance);
  }
  return 0;
}
```

The code enumerated every file in the system directory, Windows directory, Program Files directory, and possibly also the kitchen sink and their uncle's kitchen sink. It tries to load each one as a library, and sees if it has an export called `FunctionX`. For good measure, it also tries `__imp__FunctionX`, `FunctionX@12`, and `__imp__FunctionX@12`. If it finds any match, it calls the function.

As it happens, every single call to `GetProcAddress` failed. The function they were trying to call was an internal function in the window manager that wasn't exported. I guess they figured, "Hm, I can't find it in `user32`. Maybe it moved to some other DLL," and went through every DLL they could think of.

I called out this rather dubious programming technique, and word got back to the development team for that program. They came back and admitted, “Yeah, we were hoping to call that function, but couldn’t find it, and the code you found is stuff we added during debugging. We have no intention of actually shipping that code.”

Well, yeah, but still, what possesses you to try such a crazy technique, even if only for debugging?

Raymond Chen

Follow