Dubious security vulnerability: Attacking the application directory in order to fool yourself?

devblogs.microsoft.com/oldnewthing/20161013-00

October 13, 2016

Raymond Chen

A security vulnerability report arrived that went something like this:

> There is a vulnerability in the `XYZ.EXE` program. If you place a hacked copy of the file `CABINET.DLL` in the same directory as `XYZ.EXE`, then when the user runs the `XYZ.EXE` program, it loads the hacked `CABINET.DLL` instead of the real one. When `XYZ.EXE` prompts for elevation, the user will allow it, and now the rogue `CABINET.DLL` is running code as administrator.

Um, okay.

First of all, this is an application directory attack, and the application directory is considered a trusted location. If you let somebody write to your application directory, then you are giving them control over what the application does. So don't do that.

This particular variation tries to disguise the matter by throwing in an elevation prompt, but the underlying issue is the same. Let's look at it another way: Who is the attacker, and who is the victim?

The attacker is the user who creates a trap in the application directory. The victim is the person who runs the application and gets caught in the trap. But how do you get the victim to wander into the yucky hot tub? Whatever technique you used to get them to run a program from your hot tub, you can use that technique to get them to run a rogue app directly; no need for fancy application directory attacks.

The other possibility of a victim is the user himself, who runs the `XYZ.EXE` application, and discovers that he just fell into his own trap. It's not really considered elevation if you manage to fool yourself.

Raymond Chen

Follow