It rather involved being on the other side of this airtight hatchway: Open access to the application directory

devblogs.microsoft.com/oldnewthing/20130802-00

August 2, 2013

Raymond Chen

A security vulnerability report arrived claiming that the Program X installer was insecure because it loaded a DLL (let's call it `HAHA.DLL`) from the current directory, thereby being susceptible to a current directory attack. (Other terms for this type of attack are *DLL planting* and *DLL side-loading*.)

The vendors who were responsible for Program X forwarded the report to Microsoft because their program never loaded `HAHA.DLL` directly; it was being loaded by a system component.

The first order of business was to verify that it was actually a DLL planting vulnerability. And it wasn't. It was an application directory attack, not a current directory attack. It turns out that a lot of purported DLL planting vulnerability reports are actually application directory attacks. DLLs in the application directory take priority over system DLLs because the directory is the Windows equivalent of what on the Mac is called an application bundle.[1] Which only serves to highlight the importance of securing your application directory.

In the original report, Program X was in a directory called something like `\\server\software\install`, which was filled with setup programs for various applications. As a result, all of the programs were soaking in the same hot-tub.

When this issue was pointed out to the vendors of Program X, they responded, "No, this is still a bug. You need to add `HAHA.DLL` to the KnownDlls list so that it cannot be overridden by the application directory."

The KnownDlls list is not a security feature. It is a *performance* feature. The fact that KnownDlls overrides the application directory is a side-effect of its implementation (namely, to avoid directory searching for popular DLLs), and it is arguably a bug, since it breaks contractual behavior: The application directory no longer takes precedence over the system directory. The Application Compatibility folks spend a lot of time studying the KnownDlls list to make sure that the DLLs in there are ones that no properly-functioning application should be trying to override with a local copy.

Even if `HAHA.DLL` were added to the KnownDlls list, that does not guarantee that it will always be loaded from the system directory. If somebody can attack your application directory, then they can drop a DLL redirection manifest into the directory or use DotLocal DLL redirection, both of which also override KnownDlls. (Observe that both of these attacks require write access to the application directory.)

The application directory is your safety bubble. If you let anybody into your safety bubble, then it isn't very safe any more.

In the parlance of airtight hatchways: Granting open write access to your application directory is equivalent to leaving open the door to your airtight hatchway.

[1] I used to say simply "The directory is the application bundle", but I'm now forced to use the much more awkward formulation because at least one person thought I was talking about Windows Store application bundles.

Raymond Chen

Follow