**DLL Spoofing (also known as DLL preloading or a binary planting attack and Dll Hijacking)**

**Description 1**

If an attacker gains control of one of the directories on the DLL search path, it can place a malicious copy of the DLL in that directory. This is sometimes called a DLL preloading attack or a binary planting attack. If the system does not find a legitimate copy of the DLL before it searches the compromised directory, it loads the malicious DLL. If the application is running with administrator privileges, **the attacker may succeed in local privilege elevation. [1]**

**Description 2**

DLL hijacking is the practice of having a vulnerable application load a malicious library (allowing for the execution of arbitrary code), rather than the legitimate library by placing it at a preferential location as dictated by the [Dynamic-Link Library Search Order](https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx) . [2]

**Description 3**

Dynamic Link Libraries (DLL's) are software object modules, or libraries, linked into a program while it is running. DLL's are powerful features that allows for programs to share common code making them easier to develop and make them more efficient. Since the DLL code runs in the context of its host program it inherits the full capabilities of the program's user with spoofing in the general sense. The DLL spoof **causes a legitimate program** **(in the worst case scenario run by an administrator) to load a DLL with a Trojan Horse instead of the legitimate DLL. Once it gains control it has the same capabilities as the user who ran the program.**

When a program loads DLL's, it searches through a sequence of directories looking for the DLL in question. The successful spoof occurs when an attacker succeeds in inserting the malicious DLL-file in one of those directories so that the program finds it before it finds the legitimate DLL of the same name. This means that the attack can still be successful even if the legitimate DLL is beyond the attacker’s reach. [3]

**Reference**

[1] <https://msdn.microsoft.com/en-us/library/ff919712(VS.85).aspx>

[2] <https://digital-forensics.sans.org/blog/2015/03/25/detecting-dll-hijacking-on-windows/>

[3]https://www.it.uu.se/edu/course/homepage/sakdat/ht05/assignments/pm/programme/DLL\_Spoofing\_in\_Windows.pdf