HijackLibs

Enter the name of a DLL or EXE here…

What is DLL Hijacking?

DLL Hijacking is, in the broadest sense, tricking a legitimate/trusted application into loading an arbitrary DLL. Defensive measures such as AV and EDR solutions may not pick up on this activity out of the box, and allow-list applications such as AppLocker may not block the execution of the untrusted code. There are numerous examples of threat actors that have been observed to leaverage DLL Hijacking to achieve their objectives.

There are various subtypes of DLL Hijacking; this project distinguishes between the following types:

- *DLL Sideloading* (T1574.002): By copying (and optionally renaming) a vulnerable application to a user-writeable folder, alongside a malicious DLL, arbitrary code can be executed through the legitimate application.
- *Phantom DLL Hijacking*: By copying a malicious DLL to a specific location, vulnerable applications will load and execute the (normally non-existent) DLL upon normal execution.
- *DLL Search Order Hijacking* (T1574.001): DLLs specified by an application without a path are searched for in fixed locations in a specific order. By putting a malicious DLL in a location that is searched in before the actual DLL, the legitimate application will execute arbitrary code upon normal execution.
- *Environment Variable-based DLL Hijacking*: By changing a specific environment variable to an attacker-controlled directory, it is possible to trick a vulnerable application into loading a malicious from the attacker-controled location.

For an overview of useful resources explaining various aspects of DLL Hijacking, please refer to our wiki.

What is this project about?

This project provides an curated list of DLL Hijacking candidates. A mapping between DLLs and vulnerable executables is kept and can be searched via this website. Additionally, further metadata such as resources provide more context.

For defenders, this project can provide valuable information when trying to detect DLL Hijacking attempts. Although detecting DLL Hijacking isn't always without challenge, it is certainly possible to monitor for behaviour that may be indicative of abuse. To further support defenders, out-of-the-box Sigma rules are provided through this website. A σ Sigma feed containing detection rules for all entries part of this project is available too.

For red teamers, this project can help identify DLLs that can be used to achieve DLL Hijacking. The aim of this project is not to make it easy to abuse the recorded vulnerabilities; as such, PoCs, code templates or tuturials are not provided.

How can I use this project's data?

This project offers a number of generated files, including JSON and CSV, containing all of the project's data. Because they are automatically updated to the latest version, they can be used in automated processes. More information can be found on the API page.

How do I get involved?

This project is fully open source and is maintained by the community, as can be seen on our contributors page. Everyone can contribute, and new contributions are more than welcome - please head to this project's GitHub page to find the contribution guide.