# TTP Detail – T1574.006

## TTP Information

Name: Dynamic Linker Hijacking

Description: Adversaries may execute their own malicious payloads by hijacking environment variables the dynamic linker uses to load shared libraries. During the execution preparation phase of a program, the dynamic linker loads specified absolute paths of shared libraries from various environment variables and files, such as <code>LD\_PRELOAD</code> on Linux or <code>DYLD\_INSERT\_LIBRARIES</code> on macOS.(Citation: TheEvilBit DYLD\_INSERT\_LIBRARIES)(Citation: Timac DYLD\_INSERT\_LIBRARIES)(Citation: Gabilondo DYLD\_INSERT\_LIBRARIES Catalina Bypass) Libraries specified in environment variables are loaded first, taking precedence over system libraries with the same function name.(Citation: Man LD.SO)(Citation: TLDP Shared Libraries)(Citation: Apple Doco Archive Dynamic Libraries) Each platform's linker uses an extensive list of environment variables at different points in execution. These variables are often used by developers to debug binaries without needing to recompile, deconflict mapped symbols, and implement custom functions in the original library.(Citation: Baeldung LD\_PRELOAD)

Hijacking dynamic linker variables may grant access to the victim process's memory, system/network resources, and possibly elevated privileges. On Linux, adversaries may set <code>LD\_PRELOAD</code> to point to malicious libraries that match the name of legitimate libraries which are requested by a victim program, causing the operating system to load the adversary's malicious code upon execution of the victim program. For example, adversaries have used `LD\_PRELOAD` to inject a malicious library into every descendant process of the `sshd` daemon, resulting in execution under a legitimate process. When the executing sub-process calls the `execve` function, for example, the malicious library’s `execve` function is executed rather than the system function `execve` contained in the system library on disk. This allows adversaries to [Hide Artifacts](https://attack.mitre.org/techniques/T1564) from detection, as hooking system functions such as `execve` and `readdir` enables malware to scrub its own artifacts from the results of commands such as `ls`, `ldd`, `iptables`, and `dmesg`.(Citation: ESET Ebury Oct 2017)(Citation: Intezer Symbiote 2022)(Citation: Elastic Security Labs Pumakit 2024)

Hijacking dynamic linker variables may grant access to the victim process's memory, system/network resources, and possibly elevated privileges.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Kill Chain Phases

**•** mitre-attack: persistence

**•** mitre-attack: privilege-escalation

**•** mitre-attack: defense-evasion

## Malware

* COATHANGER
* Ebury
* HiddenWasp
* Hildegard
* XCSSET

## APTs (Intrusion Sets)

* APT41
* Aquatic Panda
* Rocke