# TTP Detail – T1574.004

## TTP Information

Name: Dylib Hijacking

Description: Adversaries may execute their own payloads by placing a malicious dynamic library (dylib) with an expected name in a path a victim application searches at runtime. The dynamic loader will try to find the dylibs based on the sequential order of the search paths. Paths to dylibs may be prefixed with <code>@rpath</code>, which allows developers to use relative paths to specify an array of search paths used at runtime based on the location of the executable. Additionally, if weak linking is used, such as the <code>LC\_LOAD\_WEAK\_DYLIB</code> function, an application will still execute even if an expected dylib is not present. Weak linking enables developers to run an application on multiple macOS versions as new APIs are added.  
  
Adversaries may gain execution by inserting malicious dylibs with the name of the missing dylib in the identified path.(Citation: Wardle Dylib Hijack Vulnerable Apps)(Citation: Wardle Dylib Hijacking OSX 2015)(Citation: Github EmpireProject HijackScanner)(Citation: Github EmpireProject CreateHijacker Dylib) Dylibs are loaded into an application's address space allowing the malicious dylib to inherit the application's privilege level and resources. Based on the application, this could result in privilege escalation and uninhibited network access. This method may also evade detection from security products since the execution is masked under a legitimate process.(Citation: Writing Bad Malware for OSX)(Citation: wardle artofmalware volume1)(Citation: MalwareUnicorn macOS Dylib Injection MachO)

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Kill Chain Phases

**•** mitre-attack: persistence

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

**•** mitre-attack: defense-evasion

## Tools

* Empire