Hijack Execution Flow: Path Interception by PATH Environment Variable, Sub-technique T1574.007 - Enterprise

Archived: 2026-04-05 15:18:07 UTC

Adversaries may execute their own malicious payloads by hijacking environment variables used to load libraries. The PATH environment variable contains a list of directories (User and System) that the OS searches sequentially through in search of the binary that was called from a script or the command line.

Adversaries can place a malicious program in an earlier entry in the list of directories stored in the PATH environment variable, resulting in the operating system executing the malicious binary rather than the legitimate binary when it searches sequentially through that PATH listing.

For example, on Windows if an adversary places a malicious program named "net.exe" in `C:\example path`, which by default precedes `C:\Windows\system32\net.exe` in the PATH environment variable, when "net" is executed from the command-line the `C:\example path` will be called instead of the system's legitimate executable at `C:\Windows\system32\net.exe`. Some methods of executing a program rely on the PATH environment variable to determine the locations that are searched when the path for the program is not given, such as executing programs from a Command and Scripting Interpreter.[1]

Adversaries may also directly modify the $PATH variable specifying the directories to be searched. An adversary can modify the `$PATH` variable to point to a directory they have write access. When a program using the $PATH variable is called, the OS searches the specified directory and executes the malicious binary. On macOS, this can also be performed through modifying the $HOME variable. These variables can be modified using the command-line, launchctl, Unix Shell Configuration Modification, or modifying the `/etc/paths.d` folder contents.[2][3][4]

Source: https://attack.mitre.org/techniques/T1574/007