Binary Planting | OWASP Foundation

Archived: 2026-04-05 23:17:43 UTC

Description

Binary planting is a general term for an attack where the attacker places (i.e., plants) a binary file containing malicious code to a local or remote file system in order for a vulnerable application to load and execute it.

There are various ways this attack can occur:

1. Insecure access permissions on a local directory allow a local attacker to plant the malicious binary in a trusted location. (A typical example is an application installer not properly configuring permissions on directories used to store application files.)
2. One application may be used for planting a malicious binary in another application's trusted location. (An example is the Internet Explorer - Safari blended threat vulnerability)
3. The application searches for a binary in untrusted locations, possibly on remote file systems. (A typical example is a Windows application loading a dynamic link library from the current working directory after the latter has been set to a network shared folder.)

Risk Factors

Examples

Insecure Access Permissions-based Attack

1. A Windows application installer creates a root directory (`C:\Application`) and installs the application in it, but fails to limit write access to the directory for non-privileged users.
2. Suppose the application (`C:\Application\App.exe`) loads the WININET.DLL library by calling `LoadLibrary("WININET.DLL")` . This library is expected to be found in the Windows System32 folder.
3. Local user A plants a malicious `WININET.DLL` library in `C:\Application`
4. Local user B launches the application, which loads and executes the malicious WININET.DLL` instead of the legitimate one.

Current Working Directory-based Attack

1. Suppose a Windows application loads the `DWMAPI.DLL` library by calling `LoadLibrary("DWMAPI.DLL")` . This library is expected to be found in the Windows System32 folder, but only exists on Windows Vista and Windows 7.
2. Suppose the application is associated with the `.bp` file extension.
3. The attacker sets up a network shared folder and places files `honeypot.bp` and `DWMAPI.DLL` in this folder (possibly marking the latter as hidden).
4. The attacker invites a Windows XP user to visit the shared folder with Windows Explorer.

5. When the user double-clicks on `honeypot.bp`, user's Windows Explorer sets the current working directory to the remote share and launches the application for opening the file.
6. The application tries to load `DWMAPI.DLL`, but failing to find it in the Windows system directories, it loads and executes it from the attacker's network share.

- Intranet Attacker
- Internet Attacker

- Code Injection

- Portability Flaw
- Process Control

References

- CWE-114: Process Control
- Elevation of Privilege Vulnerability in iTunes for Windows - example of Insecure Access Permissions-based Attack
- Remote Binary Planting in Apple iTunes for Windows - example of Current Working Directroy-based Attack

---