Hide Artifacts: Hidden Window, Sub-technique T1564.003 - Enterprise

Archived: 2026-04-05 14:25:48 UTC

Adversaries may use hidden windows to conceal malicious activity from the plain sight of users. In some cases, windows that would typically be displayed when an application carries out an operation can be hidden. This may be utilized by system administrators to avoid disrupting user work environments when carrying out administrative tasks.

Adversaries may abuse these functionalities to hide otherwise visible windows from users so as not to alert the user to adversary activity on the system.[1]

On macOS, the configurations for how applications run are listed in property list (plist) files. One of the tags in these files can be `apple.awt.UIElement`, which allows for Java applications to prevent the application's icon from appearing in the Dock. A common use for this is when applications run in the system tray, but don't also want to show up in the Dock.

Similarly, on Windows there are a variety of features in scripting languages, such as PowerShell, Jscript, and Visual Basic to make windows hidden. One example of this is `powershell.exe -WindowStyle Hidden`.[2]

The Windows Registry can also be edited to hide application windows from the current user. For example, by setting the `WindowPosition` subkey in the `HKEY_CURRENT_USER\Console\%SystemRoot%_System32_WindowsPowerShell_v1.0_PowerShell.exe` Registry key to a maximum value, PowerShell windows will open off screen and be hidden.[3]

In addition, Windows supports the `CreateDesktop()` API that can create a hidden desktop window with its own corresponding `explorer.exe` process.[4][5] All applications running on the hidden desktop window, such as a hidden VNC (hVNC) session,[4] will be invisible to other desktops windows.

Adversaries may also leverage cmd.exe[6] as a parent process, and then utilize a LOLBin, such as DeviceCredentialDeployment.exe,[7][8] to hide windows.

Source: https://attack.mitre.org/techniques/T1564/003