MITRE ATT&CK T1055 Process Injection

By Huseyin Can YUCEEL

Published: 2022-02-23 · Archived: 2026-04-05 17:10:44 UTC

Process injection is a powerful and widely used technique that allows adversaries to execute malicious code within the address space of a legitimate process. By injecting code into trusted processes; attackers can evade detection, escalate privileges, and maintain persistence on a compromised system. With process injection, malicious payloads can be run under the guise of legitimate applications, making it significantly harder for security tools to detect suspicious activity.

In this blog post, we explain the T1055 Process Injection technique of the MITRE ATT&CK® framework and explore how adversaries employ process injection with real-world attack examples in detail.

Adversary Use of Process Injection

Adversaries use Process Injection for various purposes, including evading detection, maintaining presence within a system, and accessing process resources such as memory and network.

It is a typical security practice to list all the processes running on a system and identify the malicious processes among the legitimate ones that are part of the operating system or installed software with recognizable names and file paths. Security mechanisms scan for processes that exhibit unusual characteristics, such as non-standard file paths or abnormal behavior, which may indicate a potential threat. Such processes are swiftly flagged as suspicious and can be killed to protect the system.

However, when adversaries embed their malicious code into an existing, trusted process, they create a challenge for detection efforts. This stealth tactic, known as **Process Injection**, allows the intrusive code to run unnoticed within the memory space of another process, making it particularly difficult for security defenses to detect and neutralize the threat.

Process injection provides two significant benefits for adversaries:

1. Privilege Escalation

If the target process has elevated privileges, the injected code will also have access to those privileges, allowing the adversary to gain greater control over the system and potentially escalate their privileges even further. For instance, if a target process has access to network resources, then the malicious code encapsulated within this process may allow an adversary to communicate over the Internet or with other computers on the same network. This privilege can enable the adversary to carry out various malicious activities, such as downloading next-stage payloads or tools, exfiltrating sensitive data, spreading malware to other systems, or launching attacks against the network.

2. Defense Evasion

Adversaries evade security controls designed to detect and block known threats by executing their malicious code under the privileges of a legitimate process. As the malicious code is hidden within the legitimate process, which is typically allow-listed, the target process acts as a camouflage for the malicious code, allowing the malicious code to evade detection and run without being noticed. Since the code is typically run directly in the memory of the legitimate process, it is difficult for disk forensics tools to detect the code, as it is not written to the disk.

Legitimate Processes Used for Process Injection

Security controls may quickly detect custom processes with unfamiliar names. Therefore, attackers use common native built-in Windows processes, such as:

- **AppLaunch.exe -** Application Launcher
- **arp.exe** – Address Resolution Protocol Utility
- **cmd.exe** – Command Prompt
- **conhost.exe** – Console Window Host
- **control.exe** – Control Panel Applet
- **csrss.exe** – Client/Server Runtime Subsystem
- **ctfmon.exe** – CTF Loader
- **dllhost.exe** – COM Surrogate
- **dwm.exe** – Desktop Window Manager
- **explorer.exe** – Windows Explorer
- **lsass.exe** – Local Security Authority Subsystem Service
- **msbuild.exe** – Microsoft Build Engine
- **mshta.exe** – Microsoft HTML Application Host
- **msiexec.exe** – Windows Installer
- **PowerShell.exe** – Windows PowerShell
- **rundll32.exe/rundll64.exe** – Run a DLL as an App
- **schtasks.exe** – Task Scheduler
- **services.exe** – Services Control Manager
- **smss.exe** – Session Manager Subsystem
- **spoolsv.exe** – Print Spooler Service
- **svchost.exe** – Service Host
- **taskhost.exe** – Host Process for Windows Tasks
- **taskmgr.exe** – Task Manager
- **wininit.exe** – Windows Start-Up Application
- **winlogon.exe** – Windows Logon Process
- **wmiexec.exe** – WMI Execution Process
- **wmiprvse.exe** – WMI Provider Host
- **wscntfy.exe** – Windows Security Center Notification App
- **wuauclt.exe** – Windows Update AutoUpdate Client

Attackers also use processes of commonly used software, such as browsers, antiviruses, office tools, and utilities.

- **acrobat.exe -** Adobe Acrobat

- **adobearm.exe** – Adobe Acrobat Reader Updater
- **chrome.exe** – Google Chrome
- **discord.exe** – Discord
- **dropbox.exe** – Dropbox
- **dropboxsync.exe** – Dropbox Sync
- **excel.exe** – Microsoft Excel
- **firefox.exe** – Mozilla Firefox
- **googleupdate.exe** – Google Updater
- **java.exe** – Java Runtime Environment
- **jucheck.exe** – Java Update Checker
- **notepad.exe** – Notepad
- **onedrive.exe** – OneDrive
- **opera.exe** – Opera Browser
- **outlook.exe** – Microsoft Outlook
- **photoshop.exe** – Adobe Photoshop
- **slack.exe** – Slack
- **steam.exe** – Steam
- **teams.exe** – Microsoft Teams
- **vmwaretray.exe** – VMware Tray
- **winword.exe** – Microsoft Word
- **wordpad.exe** – Wordpad
- **zoom.exe** – Zoom

Methods of Target Process Selection

Adversaries use the following methods when picking their target process for malicious code injection:

1. Hardcoded Targeting

In the first scenario, an adversary can hardcode a particular target process in the malicious code, and only this process is used to host the injected code. explorer.exe and rundll32.exe are the two most commonly leveraged processes for this type of attack. For instance, RedLine Stealer malware is known to target the Visual Basic Compiler used with the .NET Framework. The malware injects its payload into the vbc.exe to evade detection [1].

An attacker can also define a list of target processes in the code, and the injected code is executed in the first process on the list that is found to be running on the system. These lists typically include native Windows and browser processes.

2. Dynamic Targeting

In this attack scenario, an adversary does not define the target process beforehand and instead locates a suitable host process at runtime. It is common for adversaries to use Windows API functions to enumerate the list of all currently active processes and to get a handle on each target process in attack campaigns. The specific API functions that are used will depend on the goals of the attack and the capabilities of the adversary, but some

common examples include EnumProcesses(), EnumProcessModules(), CreateToolhelp32Snapshot(), and OpenProcess().

Sub-techniques of T1055 Process Injection

Process injection is not a single technique but a collection of subtechniques that adversaries use to execute malicious code within legitimate processes. In version 16.1, MITRE ATT&CK Matrix for Enterprise has 12 subtechniques under T1055 Process Injection, each with unique characteristics and attack scenarios. These subtechniques vary in complexity and effectiveness, but they all serve the same goal, stealthy execution and evasion from security defenses.

For more detailed information, check out the following blogs explaining each subtechnique in great detail.

- T1055.001 Dynamic-link Library Injection
- T1055.002 Portable Executable Injection
- T1055.003 Thread Execution Hijacking
- T1055.004 Asynchronous Procedure Call
- T1055.005 Thread Local Storage
- T1055.008 Ptrace System Calls
- T1055.009 Proc Memory
- T1055.011 Extra Window Memory Injection
- T1055.012 Process Hollowing
- T1055.013 Process Doppelgänging
- T1055.014 VDSO Hijacking
- T1055.015 ListPlanting

Ready to Simulate Real-World Threats From Red Report 2026?

References

[1] S. Gandy, "RedLine Stealer Malware Analysis," Cyber Florida: The Florida Center for Cybersecurity, Mar. 10, 2023. Available: https://cyberflorida.org/redline-stealer-malware-analysis/
