Process Injection: Proc Memory, Sub-technique T1055.009 - Enterprise

Archived: 2026-04-02 12:35:20 UTC

Adversaries may inject malicious code into processes via the /proc filesystem in order to evade process-based defenses as well as possibly elevate privileges. Proc memory injection is a method of executing arbitrary code in the address space of a separate live process.

Proc memory injection involves enumerating the memory of a process via the /proc filesystem (`/proc/[pid]`) then crafting a return-oriented programming (ROP) payload with available gadgets/instructions. Each running process has its own directory, which includes memory mappings. Proc memory injection is commonly performed by overwriting the target processes' stack using memory mappings provided by the /proc filesystem. This information can be used to enumerate offsets (including the stack) and gadgets (or instructions within the program that can be used to build a malicious payload) otherwise hidden by process memory protections such as address space layout randomization (ASLR). Once enumerated, the target processes' memory map within `/proc/[pid]/maps` can be overwritten using dd.[1][2][3]

Other techniques such as Dynamic Linker Hijacking may be used to populate a target process with more available gadgets. Similar to Process Hollowing, proc memory injection may target child processes (such as a backgrounded copy of sleep).[2]

Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via proc memory injection may also evade detection from security products since the execution is masked under a legitimate process.

Source: https://attack.mitre.org/techniques/T1055/009