Process Injection: Ptrace System Calls, Sub-technique T1631.001 - Mobile

Archived: 2026-04-05 16:08:06 UTC

Adversaries may inject malicious code into processes via ptrace (process trace) system calls in order to evade process-based defenses as well as possibly elevate privileges. Ptrace system call injection is a method of executing arbitrary code in the address space of a separate live process.

Ptrace system call injection involves attaching to and modifying a running process. The ptrace system call enables a debugging process to observe and control another process (and each individual thread), including changing memory and register values.[1] Ptrace system call injection is commonly performed by writing arbitrary code into a running process (e.g., by using `malloc`) then invoking that memory with `PTRACE_SETREGS` to set the register containing the next instruction to execute. Ptrace system call injection can also be done with `PTRACE_POKETEXT` / `PTRACE_POKEDATA`, which copy data to a specific address in the target process's memory (e.g., the current address of the next instruction).[1][2]

Ptrace system call injection may not be possible when targeting processes with high-privileges, and on some systems those that are non-child processes.[3]

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 ptrace system call injection may also evade detection from security products since the execution is masked under a legitimate process.

Source: https://attack.mitre.org/techniques/T1631/001