Process Injection

Created by Kyle Nehman, last modified on Oct 19, 2022

What are process injections?

Process injections are when a (potentially malicious) process A forces process B to execute instructions that are not otherwise a part of its code. Process injection has many different techniques, the most common of which being to create a remote thread inside of the victim process.

Often times, these new instruction are created side by side with the original program, so normal operations of the victim process B are not affected. Because of this, the injection will often go unnoticed.

Relevant Mitre tactics:

- Mitre T1055 - Process injection, as described above. Used to evade malware detection by allowing the target process to continue operating normally while executing malicious code.
- Mitre T1559 - Inter-Process communication can provide control over the target process from the injector once the injection is complete.
- Mitre T1569 - Injecting into a system service such as an existing `svchost` can disguise malicious code to be reported as running from a well known and trusted process.

What are the consequences?

Untrusted process injections into an otherwise trusted process means that it’s no longer trusted. An otherwise uncorrupted installation of secured software, like a SQL server, can be forced to execute malware payloads. After process death and/or reboot of the system, any evidence of this tampering will be gone, since everything was purely done in memory with dynamic thread creation.

What does Cyber Crucible do?

Cyber Crucible monitors process injections that occur on the system, and does not stop the activity unless data extortion activities or ransomware encryption have begun. This is because some processes on the system will communicate via different process injection techniques, and will not

perform any malicious behavior.

Instead, Cyber Crucible will keep track of the injections that happen to and from each process on the system, and record it for use with potential root cause analysis later. This way we can know not just that a particular process is currently untrusted in memory, but also which process started the chain of malicious activity.

How do I tell?

31784977.png?width=510
When an automated response has the “untrusted remote threads” field true, we know we’re dealing with process injections. The good news is that we can grab the pid of the response, and search through the process injections around the time of the incident, and find the perpetrator.

Injector Pid	Injector Path	Injectee Pid	Injectee Path ↑
3992	C:\Windows\System32\csrss.exe	8844	C:\Windows\System32\cmd.exe
4228	C:\Windows\System32\dwm.exe	3992	C:\Windows\System32\csrss.exe
3268	C:\Windows\System32\dwm.exe	5972	C:\Windows\System32\csrss.exe
4228	C:\Windows\System32\dwm.exe	3992	C:\Windows\System32\csrss.exe

Example Scenarios:

- Training Scenario - Process Injection

Revision #2
Created 2026-05-18 20:29:23 UTC by Dennis Underwood
Updated 2026-05-18 20:40:44 UTC by Dennis Underwood