ATT&CK® Deep Dive: Process Injection - Red Canary

By Share

Archived: 2026-04-05 12:57:13 UTC

WHY FOCUS ON PROCESS INJECTION?

Process Injection encompasses a wide array of malicious behaviors that offer adversaries an inconspicuous method of evading defensive controls, elevating their privilege level, or otherwise executing arbitrary code. It's so broad that in the next ATT&CK release, MITRE is recategorizing the technique into 11 sub-techniques.

As such, this is the perfect time for an in-depth, technical conversation exploring the ways that adversaries leverage Process Injection, what malicious process injection looks like, and how you can detect it.

See highlight clips from the Process Injection webinar.

01:25 Panelist Introduction

02:10 What Red Canary Does

02:35 Process Injection Definition

02:48 "Process injection is a way of running arbitrary code in another process's memory space." – Adam

03:30 Why Leverage Process Injection

03:50 "It can be a very good way to evade defenders and defensive controls that are focused around specific tools." – Adam

04:55 Sub-techniques of Process Injection

05:20 "Process injection, which is a fairly broad technique, has been broken out into 11 different sub-techniques which are more specific techniques or ways of actually doing process injection." – Adam

05:45 Webinar Agenda

07:11 Portable Executable Injection

08:09 "The ability to inject that into another process and then invoking it by spawning a new thread." – Matt

09:40 "This technique offers an attacker particular flexibility in that it facilitates direct injection into their target." – Matt

10:57 Ramnit

11:15 "In the case of Ramnit, it injects a traditional DLL into a browser process." – Matt

13:55 Observing PE Injection

14:14 "What are the absolute minimum requirements an attacker has imposed in order to successfully carry out the attack technique?" – Matt

17:05 Detecting PE Injection

18:15 "Since they're already writing their own position-independent code, there's no hard requirement that they must write a PE header into the memory space of another process, but a lot of malware does." – Matt

21:30 Thread Local Storage

22:29 "Although it is kind of uncommon, it can be a successful anti-analysis technique because typically analysis is going to start at the entry point of the PE, and this malware is going to get executed before that is reached." – Erika

22:52 Ursnif

23:05 "It's often spread via malicious Office documents" – Erika

23:23 Observing Malicious TLS Callbacks

23:29 "Within the PE header itself, there is a TLS directory which contains information about TLS data objects. These thread local storage objects basically allow each thread to have its own static data area for custom variables, or custom thread initialization routines that they want to employ." – Erika

28:29 Detecting Malicious TLS Callbacks

29:22 "We talked about some of the APIs that are used by this technique, you can obviously detect the usage of those, but the usage of any one of those by itself is not going to indicate malicious activity." – Erika

30:38 Process Hollowing

31:00 "It allows an attacker to carve out the executable section of a legitimate or benign process and replace it with their own payload." – David

33:42 Process Hollowing: Generalized Technique

36:05 "Once the attacker has obtained the PE address, the third step is to perform the hollowing itself." – David

41:55 Trickbot

42:32 "What's interesting for us as technical defenders is that we use a number of varied techniques taking shape." – David

43:52 Process Hollowing: Trickbot

45:30 "They create a section that describes their unpacked code. This is created within the originator process, meaning within the malware process itself. They create a new section that describes the unpacked code to inject or

make resident within the hollowed process.” – David

47:52 Detecting Process Hollowing

51:12 “It’s difficult to get a robust detection at scale.” – David

52:16 Linux

57:09 “We only had these three Linux-only techniques in ATT&CK today.” – Adam

57:26 MacOS

57:38 “Injection isn’t seen in malware on the Mac platform, and there are a few reasons for that. The most prevalent being that it is much harder than getting the user to click on something and type in their password.” – Erika

01:00:52 Mitigating Process Injection

1:01:02 “There is no specific mitigation against all forms of process injection.” – Matt

01:04:05 Questions & Answers

We will be following up with questions and answers in a blog post soon.
