Training Scenario - Process Injection

Group Name

Training Data - Process Injection (Hive)

Scenario

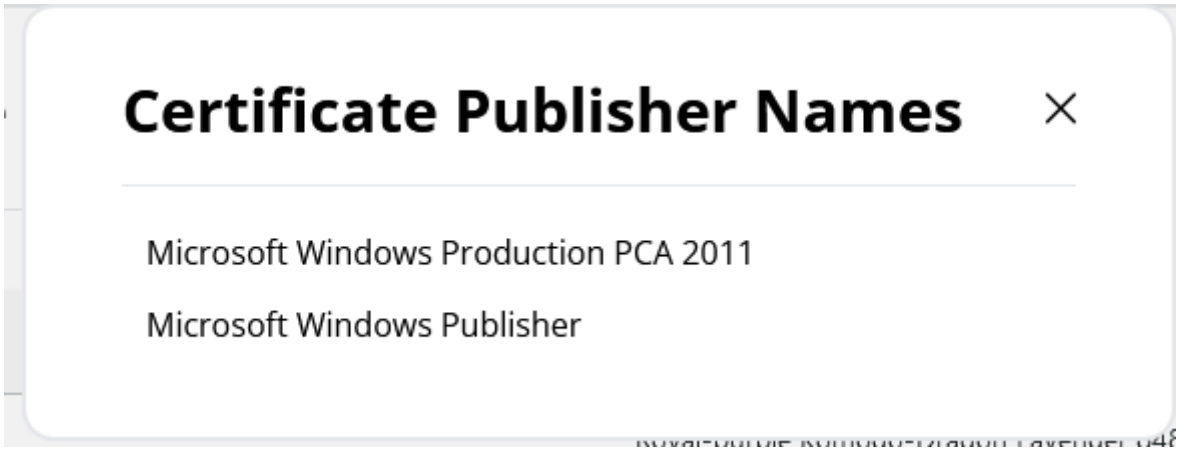
In this training scenario, we will execute a ransomware payload via process injection into an already running Svchost.exe. The Svchost in question is a normal system operation, long running, signed, and under the LocalSystem user account.

Because the Svchost is ‘real’ it is often overlooked. Admins know that many instances of Svchost are run, and as long as the program’s arguments look correct, it is treated as a “black box” and left alone. This is an obvious, and often exploited, loophole for hackers to take advantage of. Combined with a zero day payload, this approach is very successful at obfuscating ransomware attacks.

Identifying it

Number of Incidents	Machine Name	Program Path	Program Arguments
1	\\CC-NOAH-TEST	C:\Windows\System32\svchost.exe	-k UnistackSvcGroup

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 9:16:41 PM EDT	Royal-purple Komc	True	False	False



svchost.exe's Cert Signer(s)

The situation around this automated response is not directly apparent at a glance. The command line arguments look valid for Svchost, and the executable is signed by Microsoft.

The fact that there is an untrusted remote thread is suspicious, but requires more digging into. Many remote threads happen on a system, but most of them remain "trusted" as they are used for normal inter-process communication, data sharing, etc.

When a remote thread is created into a process in an abnormal way, and is marked as untrusted, is where we need to scrutinize harder.

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

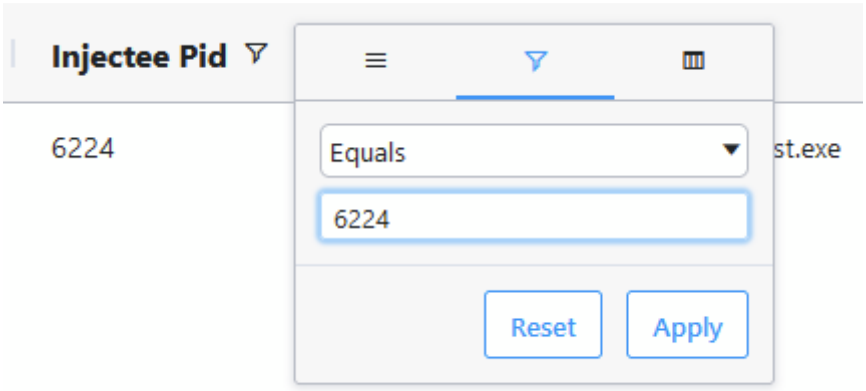
Innocent Process Injections that occur normally

In order to sort through the many process injections that happen on a system at any given time, we can refer to both the time of the incident, and the pid of the flagged process. Normally we will start with a few pids of suspicious responses, and go up the chain of injections and creation to gain the full story.

Response Details

Description	Value
Response Name	Royal-purple Komodo-Dragon Lavender 848
Machine Name	\\CC-NOAH-TEST
Program Path	C:\Windows\System32\svchost.exe
Program Arguments	-k UnistackSvcGroup
Agent Name	
Ram usage	0
File access count	0
Number of threads	0
Pid	6224

View Certificate Publisher Names



Filtering out irrelevant injections

Injector Pid	Injector Path	Injectee Pid	Injectee Path
9168	C:\svchost_injector.exe	6224	C:\Windows\System32\svchost.exe

The smoking gun!

We already know that no one should be injecting into Svchost in this way, but definitely not this process! From here on we can treat this process as if it's "malware.exe" and see who ran it, since someone must have.

Created ↑	Child Pid ▽	Parent Path	Child Path
October 18, 2022 at 9:11:09 PM EDT	9168	C:\Windows\System32\services.exe	C:\Program Files (x86)\Dropbox\Update\DropboxUpdate.exe
October 18, 2022 at 9:16:04 PM EDT	9168	C:\Windows\explorer.exe	C:\svchost_injector.exe

Processes started with pid 9168

Since there were two process creations with pid 9168, and they are very different, we can narrow it down based on the timestamp and/or the child path, and see that it was created by an explorer.exe. This means someone manually ran the injector process, and it can't be some coincidence of Svchost behavior that was incorrectly flagged.

Relevant documentation

- **Process Injection**
- Mitre T1559 - Inter-Process communication can provide control over the target process from the injector once the injection is complete.
- Mitre T1106 - Native APIs are often the closest access to operating system functionality for accessing files, running processes, and more.
- 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.
- Mitre T1055 - Process injection, usually used to run malicious code in a target process while allowing the original process to continue.

Revision #2
Created 2026-07-13 18:49:01 UTC by Dennis Underwood
Updated 2026-07-13 18:49:07 UTC by Dennis Underwood