




Blog Post

Home / DFIR Research / The Key to Identify PsExec

The Key to Identify PsExec



By Devon On January 18, 2023

Summary:

In one way or another, PsExec – a wildly popular remote administration tool in the [Microsoft SysInternals Suite](#) – peeks its head in the wild. Threat actors tend to leverage PsExec for various reasons, such as executing commands or programs on a remote host in a victim’s environment, or for more nefarious reasons, such as deploying ransomware. The focus of this blog is to bring attention to a relatively new method of identifying the source host from which PsExec was executed from. Huge shoutout to Joe Ziemba for bringing this to my attention on a ransomware case we worked on together!

Target Host Example:

There are a few ways to identify PsExec was executed on a target system, but I will focus on the ever-so-fruitful System 7045 (Service Install) event. Any time PsExec is executed on a target host, a forensic analyst will stumble on a System 7045 event for PSEXESVC at this location: %SystemRoot%\PSEXESVC.exe. PSEXESVC is the service that gets installed on the destination host which was on the receiving end of a PsExec command. Keep in mind, PSEXESVC will be the default service name on a target system any time PsExec is executed but can be changed using the -r option which is used to specify the name of the remote service to create or interact with.

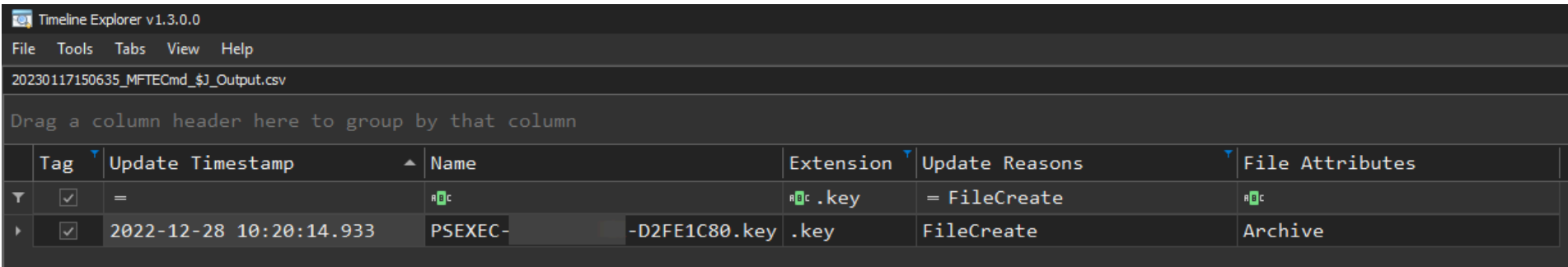
Source Host Example:

One method to identify the source system from which a PsExec command was remotely executed from is the Security 4624 event that everyone is so fond of – specifically Type 3 Network Logons. Anytime PsExec is executed on a target system, a 4624 Type 3 Logon will be generated on the target. An analyst can get lucky and cross-correlate the Type 3 Logon time with a service install for PSEXESVC which should be off by a few milliseconds.

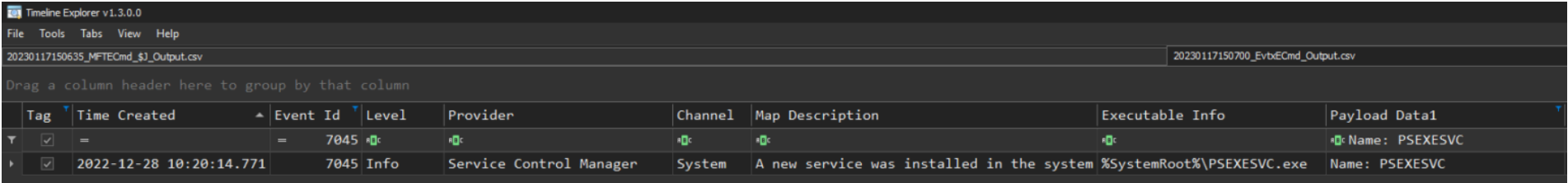
PsExec Key File (New Identification Method – USN Journal):

Starting with PsExec v2.30 (which was released in early 2021), anytime a PsExec command is executed, a .key file gets written to the file system and will be recorded in the USN Journal on the target system. It will follow this naming convention: PSEXEC-[Source Hostname]-[8 Unique Characters].key and will be located at the c:\windows directory. Keep in mind, if PsExec is run on a system against itself, the .key filename will include the same source hostname as the system in which PsExec was run on (i.e. Running PsExec from host Win10VM against itself will generate a .key file in the USN Journal on Win10VM with the following naming convention: PSEXEC-Win10VM-[8 unique characters].key).

See the below screenshot of the PSEXEC .key file being recorded in the USN Journal as a FileCreate event using output that was generated by Eric Zimmerman’s [MFTECmd.exe](#) and loaded into [Timeline Explorer](#) (Note – Client data was redacted for confidentiality):



Now when you stack the USN Journal FileCreate event for the PSEXEC .key file with a System 7045 event, you can see the timestamps are off by a few milliseconds. See the below screenshot of the System 7045 event using output that was generated by Eric Zimmerman’s [EvtxECmd.exe](#) and loaded into [Timeline Explorer](#) (Note – Client data was redacted for confidentiality):



This was found on a recent ransomware engagement I worked on which allowed my team and I to track most of the PsExec activity the threat actors performed in this environment. This can prove highly beneficial to properly understand what systems were compromised by the threat actors, especially during a ransomware incident in which threat actors tend to leverage PsExec to deploy ransomware across a victim’s environment.

UPDATE – PsExec Key File (New Identification Method – Prefetch):

Huge thanks to Richard Davis for discovering another way to identify the PSEXEC .key files using Prefetch. He demonstrates this in his 13Cubed video “Detecting PsExec Usage” which will be linked below in the article.

Prefetch, which is an evidence of execution artifact in Windows, can also be leveraged to identify PSEXEC .key files similarly to the USN Journal. When PsExec is executed, a PSEXESVC.exe Prefetch file will be created on the target system by default (or whatever service name was supplied in the PsExec command that was executed from the source system with the -r option I mentioned earlier). This Prefetch file can then be parsed in order to find all of the files that were referenced by it’s execution. In the screenshot below, I parsed a PSEXESVC.exe Prefetch file that was generated on the target system after PsExec was executed from the source system using Eric Zimmerman’s [PECmd.exe](#) and loaded into [Timeline Explorer](#) (Note – Client data was redacted for confidentiality):

```
Command line: -f C:\Users\fabianmendoza\Documents\DFIR\PSEXESVC.EXE-7F956DAF.pf

Warning: Administrator privileges not found!

Keywords: temp, tmp

Processing C:\Users\fabianmendoza\Documents\DFIR\PSEXESVC.EXE-7F956DAF.pf

Created on: 2023-07-05 22:46:20
Modified on: 2023-07-05 22:46:20
Last accessed on: 2023-07-05 22:55:12

Executable name: PSEXESVC.EXE
Hash: 7F956DAF
File size (bytes): 25,194
Version: Windows 10 or Windows 11

Run count: 10
Last run: 2022-12-28 11:15:16
Other run times: 2022-12-28 11:02:54, 2022-12-28 10:53:19, 2022-12-28 10:46:01, 2022-12-28 10:39:07, 2022-12-28 10:38:59, 2022-12-28 10:20:29, 2022-12-28 10:15:42

Volume information:

#0: Name: \VOLUME{01d46679cae64375-86cb17b3} Serial: 86CB17B3 Created: 2018-10-18 00:30:38 Directories: 11 File references: 56
```



```
Command Prompt

C:\Users\fabianmendoza\Documents\DFIR\Tools\Zimmerman Tools>PECmd.exe -f "C:\Users\fabianmendoza\Documents\DFIR\PSEXESVC.EXE-7F956DAF.pf" | findstr .KEY
36: \VOLUME{01d46679cae64375-86cb17b3}\WINDOWS\PSEXEC-[REDACTED]-3774BDE7.KEY
45: \VOLUME{01d46679cae64375-86cb17b3}\WINDOWS\PSEXEC-[REDACTED]-BCC343E1.KEY
46: \VOLUME{01d46679cae64375-86cb17b3}\WINDOWS\PSEXEC-[REDACTED]-2295E46D.KEY
50: \VOLUME{01d46679cae64375-86cb17b3}\WINDOWS\PSEXEC-[REDACTED]-0C175B1C.KEY
51: \VOLUME{01d46679cae64375-86cb17b3}\WINDOWS\PSEXEC-[REDACTED]-6041356B.KEY
53: \VOLUME{01d46679cae64375-86cb17b3}\WINDOWS\PSEXEC-[REDACTED]-8B414997.KEY

C:\Users\fabianmendoza\Documents\DFIR\Tools\Zimmerman Tools>|
```

As you can see, after parsing the PSEXESVC.exe Prefetch file from the target system, we were able to identify six PSEXEC .key files which show us the source hostnames in which PsExec was executed from – similarly to how the USN Journal does!

Conclusion:

More research will need to be conducted on why at times when a System 7045 event exists for PSEXESVC.exe on a target system, there isn’t a correlating FileCreate event being recorded in the USN Journal for the PSEXEC .key file. Still, I found these PSEXEC .key files to be very beneficial and adds another quick way for forensic analysts to identify PsExec in the wild when threat actors inevitably leverage it.

Resources:

For reference on the various ways to identify a source system that executed PsExec, check out the widely known [SANS FOR508 Hunt Evil Poster](#).

Check out this community post by Microsoft that initially mentioned this feature change with PsExec v2.30: <https://techcommunity.microsoft.com/t5/sysinternals-blog/sysmon-v13-01-and-psexec-v2-30/ba-p/2054904>.

Check out Richard Davis’ 13Cubed video on [Detecting PsExec Usage](#).

CATEGORIES DFIR Research

TAGS Prefetch PsExec USN Journal Windows artifacts

SHARE



Related Posts



Day 3 – Locard’s Exchange Principle and #DFIR



New Windows 11 Pro (22H2) Evidence of Execution Artifact!



DFIR FYI: Security:4624 has been updated in Windows 11 Pro (22H2)

