

# PSEXec as a Helper in Troubleshooting

## What

The PS Tools from Microsoft (from SysInternals) are a terrifically powerful tool to help you troubleshoot when all else fails. In this article we'll look at how you can use PSEXec to help troubleshoot an ill-behaving FileWave Client.

## When/Why

From time to time, things don't work right. None of us would be employed if this weren't the case, so let's look on the bright-side of that! But what to do if a FileWave client on a Windows device is misbehaving, and you can't communicate through normal FileWave channels? PSEXEC to the rescue.

## How

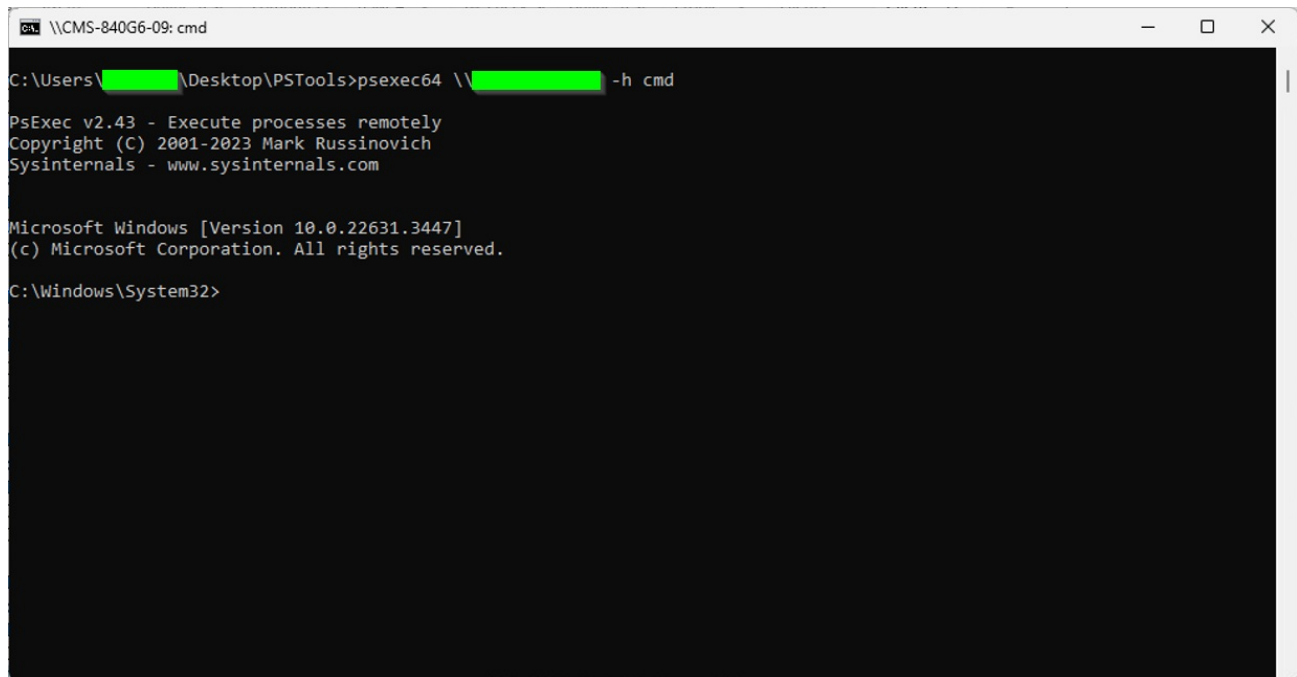
Assumptions made in the following:

- 1) You have download PSTools, and unzipped
- 2) That you launched a cmd prompt as a domain admin user (makes credentials issue easier to deal with)
- 3) That you have changed directory into the directory where PSTools is located

We'll start by simply connecting to the remote computer by name in an interactive PSEXEC shell:

```
psexec64 \\computername -h cmd
```

You'll end up in a shell like the below ('exit' will allow you to leave that shell)



```
\\CMS-840G6-09: cmd

C:\Users\████████\Desktop\PSTools>psexec64 \\████████ -h cmd

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>
```

Now, what's remarkable about this is that shell is running as your domain admin account, and you can do anything on it you can do from the command line. This article isn't meant to be a Windows CLI primer, but the following are some examples of things we could do if we assume we have a device that isn't reporting in correctly:

1. Check the FileWave Client Service:

- ```
sc query filewavewinclient
```

```
SERVICE_NAME: filewavewinclient
        TYPE                : 10  WIN32_OWN_PROCESS
        STATE                 : 4   RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE        : 0   (0x0)
        SERVICE_EXIT_CODE     : 0   (0x0)
        CHECKPOINT             : 0x0
        WAIT_HINT              : 0x0
```

2. Stop the FW Client Service

- `sc stop filewavewinclient`

3. Restart the FW Client Service

- `sc start filewavewinclient`

4. If the service won't start or stop, maybe we need to kill it forcefully by:


1. Looking for the client process

- `tasklist | findstr fwcl`

```
C:\Windows\System32>tasklist | findstr fwcl
fwcl.exe                16264 Services                0      29,660 K
```

2. And then killing it by PID

- `taskkill /PID 16264 /F`

 Note that this same procedure can be very helpful to clear up a misbehaving Windows Update agent. When Windows Update hangs, the service itself usually won't stop. Taskkill /SVC | find wuauclt will identify the proper task to stop to correct this. (A reboot is also corrective for this, but obviously impacts the use of the device)

5. Check the FW Client Log for entries from today

- `type c:\programdata\filewave\fwclient\fwcl.log | findstr mm-dd`

- (where mm-dd is today's date such as 05-16)

6. Get the IP of the workstation

- `ipconfig`

7. Restart the device (which is obviously destructive to any existing user)

- `shutdown -r -t 0 -f`

8. Determine if there are other users logged in

- `quser`

```
C:\Windows\System32>quser
USERNAME          SESSIONNAME      ID  STATE  IDLE TIME  LOGON TIME
bctowery          console         2   Active    none      5/14/2024 12:04 PM
```

9. Get the last boot time

- `Get-CimInstance -ClassName Win32_OperatingSystem | Select-Object LastBootUpTime`

```
C:\Windows\System32>wmic path win32_operatingsystem get lastbootuptime
LastBootUpTime
20231208021432.923243-300
```

## Related Content

- [PS Tools](#)
- [Great reference for Windows CLI commands \(SS64\)](#)
- [wmic deprecation](#)

