TROUBLESHOOTING WINDOWS PROBLEMS WITH POWERSHELL

Guy Leech (@guyrleech)

PowerShell User Group Rhein-Neckar, February 2023

GUY LEECH

- Independent consultant, developer, trainer, troubleshooter, comedian
- Citrix CTP, Microsoft MVP, VMware vExpert, Parallels VIPP
- @guyrleech
- guyrleech.wordpress.com
- github.com/guyrleech
- pastebin.com/u/guyrleech
- www.youtube.com/c/GuyLeech42
- linkedin.com/in/guyrleech/
- Available for hire

IN PERSON POWERSHELL WORKSHOPS

WHY USE POWERSHELL FOR TROUBLESHOOTING?

- Consistency
- Speed
- Time of day (or night) – automation via scheduled tasks
- Lazy
  - Someone probably has done what you want already (but check script!)
  - Better things to do than do admin stuff
- Improve skills
- Remediate/fix too
- Easier to share
- Log analysis with regular expressions (regex)
- Less comeback

WHAT SORT OF ISSUES?

- Processes
- Services
- Permissions
- Network
- Slow Logons
- Certificates
- Registry
- SQL
- Active Directory
- …..

WMI/CIM

- Huge amount of available information (over 800 non performance classes by default)
- Tab completion of classes or list with Get-CimClass (PoSH v3+)
- Great way to get computer details and export to CSV for reference/analysis
- Some classes have methods which can be called, e.g. Win32_UserProfile
- Filter in query, not afterwards if possible (speed,memory)
- Can take array of machines via –ComputerName
- No WMI calls in pwsh 7.x – use CIM
- Other name spaces, e.g. SCCM, Citrix
  - Get-CimInstance -Namespace Root -ClassName __Namespace
  - Tab/control space completion so easy to explore

SOME USEFUL WMI/CIM CLASSES

- Win32_Process
  - Gives parent process details which Get-Process doesn't
  - Need to invoke GetOwner method to get owner via Invoke-CimMethod
  - If on multi-user OS, filter by SessionId if relevant
- Win32_OperatingSystem
  - LastBootUpTime
- Win32_LogonSession & Win32_LoggedOnUser
  - Gives precise logon times for all logons since boot
- Win32_ComputerSystem
- Win32_Service
  - Executable including path which Get-Service doesn't
- Win32_SystemDriver
- But don't use Win32_Product as it isn't passive
  - Interrogate the registry
- Script to gather data via CIM for health checks, spot the difference, etc

QUERYING EVENT LOGS

- There are over 300+ - how many have you been looking at?
- `Get-WinEvent -ListLog * | ? IsEnabled` (408 on my Win11 laptop)
- `Get-WinEvent -ListProvider *` (1255 on my Win11 laptop)
- Filter left for speed (hashtable, XPath or XML)
  - Hashtable can filter on event id, provider, log name, start & end times, level & more
  - `Get-WinEvent -FilterHashtable @{ LogName = 'Security' ; ID = 4688 ; StartTime = '17:00' ; EndTime = '17:20' }`
- Filter/select on Properties array rather than entire message
  - `Get-WinEvent -FilterHashtable @{ LogName = 'Security' ; ID = 4688 ; StartTime = '17:00' ; EndTime = '18:00' }|Where { $_.Properties[5].Value -match '\\cmd\.exe' }`
- Much easier to visualise with Out-GridView than eventvwr
  - Can then filter in/out
  - Or save via Export-CSV
- Can be remoted so don't need to logon
- Script to bring all event logs together (was a one liner)

GET-WINEVENT

- `Get-WinEvent -ListLog *terminalservices*`
- `Get-WinEvent -LogName Application`
- `Get-WinEvent -ProviderName 'Application Error' | Select-Object -First 10`
- Filtering
  - Don't filter afterwards (`Where-Object` or `.Where()`) if possible for speed
  - Uses XML, XPath or Hash tables (dictionaries)

```
Get-WinEvent -FilterHashtable @{ LogName = 'Security' ; ID = 4688 ; StartTime = '17:00' ;
EndTime = '17:20' }
```

```
Get-WinEvent -ListLog *|? RecordCount |%{ Get-WinEvent -EA Silent -FilterHashtable
@{logname=$_.logname;starttime='16:29';endtime='16:31'}}|select *|sort TimeCreated|ogv
```

  - Filter/select on Properties array rather than entire message

```
Get-WinEvent -FilterHashtable @{ LogName = 'Security' ; ID = 4688 ; StartTime = '17:00' ;
EndTime = '18:00' }|Where { $_.Properties[5].Value -match '\\cmd\.exe' }
```

  - Script to query all event logs in a given period

PROCESS AUDITING

- Enable creation & termination success auditing
  - GPO
  - Secpol.msc (Advanced Audit Policy Configuration->Detailed Tracking)
  - auditpol.exe (use GUIDs if non-English locale)
  - P/Invoke AuditSetSystemPolicy
  - Also cmd line auditing (potential security implications)
  - Increase Security event log size/make persistent
- Event ids 4688 and 4689 in Security event log
- Can troubleshoot as far back as the Security event log goes
  - Did something crash or exit prematurely (and if so what exit code)?
  - What launched that cmd.exe at logon?
  - Is something running a lot?
- Script to show processes, durations, parents, command lines or summary

REMOTING

- Many cmdlets take –ComputerName and array of computers (comp1,comp2)
- Invoke-Command
- Winrm quickconfig
- Enter-PSSession
  - Similar to telnet/ssh access
  - Less resource intensive way to get access to troubled system
  - No GUI programs
  - Great for running SysInternals procmon headless, e.g. Windows 10
  - Accessing UNCs requires extra configuration
- UNC access can be challenging within remoted commands/sessions

POWERSHELL (PS) DRIVES

- Use standard cmdlets like Get-ChildItem, Remove-Item with various data sources
  - File system
  - Registry
  - Certificates
  - Variables
- Tab Completion
- Not all operations/properties implemented by all providers
- Additional PS providers
  - SQLPS (Use .NET SQL provider => one less dependency/pre-requsite)
  - VMware Datastores (can be slow, consider read-only /folder web interface)
  - Citrix CVAD datastore (generic interface to VMs & snapshots regardless of Hypervisor)

ACTIVE DIRECTORY

- Install ActiveDirectory module or use built-in ADSI or WMI/CIM
- Great for bulk/complex queries and changes
- Report to csv/html and email via scheduled task, e.g. expired/expiring accounts
- Do big lookups and cache (hashtable) rather than lots of individual requests
- Be careful!
  - Use `-WhatIf/-Confirm (SupportsShouldProcess/ConfirmImpact/ShouldProcess)`
  - Backup/test in non-production
- GroupPolicy module
  - E.g. what GPOs have changed in the last 7 days?

GUY'S TOP 10 TROUBLESHOOTING CMDLETS

1. Get-CIMInstance
2. Get-WinEvent
3. Enter-PSSession (etsn)
4. Out-GridView (ogv)
5. Export-CSV (epcsv)
6. Test-NetConnection (tnc)
7. Get-ADUser
8. Get-ChildItem (dir, gci, ls)
9. Get-Process (ps, gps)
10. Get-Command (gcm)

(In no particular order)

BUBBLING UNDER

- AppVClient module
- Get-Counter – get any performance counter
- Get-AuthenticodeSignature – is file signed?
- Set-ACL – fix permissions, copy from a known good system
- Get-FileHash – are those two files the same? Is that download ok?
- *-Service – start/stop services or change settings
- Stop-Computer/Restart-Computer – errors if anyone else logged on unless –force
- Invoke-WebRequest – is the web site working?

EXAMPLE USAGE (1)

- Check port open (telnet.exe equivalent, ping can be too basic)

```
Test-NetConnection 192.168.0.4 -Port 443
```

- Show expiring certificates

```
dir Cert:\LocalMachine\Root|? NotAfter -lt (Get-Date).AddDays( 90 ) |select subject,notafter
```

- Show a specific process' CPU usage (no GUI)

```
Get-Date;ps -name tiworker|select -exp TotalProcessor*|select -exp TotalSeconds
```

- Show overall CPU usage (no GUI)

```
Get-Counter -counter "\Processor(_Total)\% Processor Time"
```

- Count registry keys (registry bloat issue giving slow logon)

```
dir  "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy" -Recurse|measure
```

- Show all Citrix processes

```
ps |? Path -match 'Citrix'
```

EXAMPLE USAGE (2)

- When did that process/service start?
  ```
  ps -name blah | Select id,starttime
  ```
- Searching for files (for content)
  ```
  dir searchfolder\*.xml -Force -Recurse|sls 'searchstring|regex'
  ```
- What version are those files?
  ```
  dir searchfolder\*.exe |select -expand VersionInfo
  ```
- Show executable path & version info of a running process
  ```
  ps -name process_name|gp -ea si|select -Expand VersionInfo
  ```
- Show all McAfee services
  ```
  Get-Service | ? DisplayName -match 'mcafee'
  ```
- Diagnose IIS/Web app issues via IIS logs

ONE LINERS

- Overrated as makes understanding difficult but can be useful – copy'n'paste

```
gc logfile|? { $_ -match '^(\d|#Fields)' } | %{ $_ -replace '^#Fields: ' }|ConvertFrom-Csv -Del ' '|select *,@{n='Duration';e={([int]$_.'time-taken')}}|ogv
```

```
Get-WinEvent -ListLog * |?{ $_.RecordCount }|%{ Get-WinEvent -ea SilentlyContinue -FilterH @{logname=$_.logname;starttime='16:29:15';endtime='16:31:15'}}|select *|sort TimeCreated|Out-GridView
```

```
dir "C:\path" -force -Rec|?{ $_.PSIsContainer }|%{ if( ( Compare-Object ($acl = Get-Acl $_.FullName) (Get-Acl ($remote=$_.FullName -replace '^([A-Z]):' , '\\machine2\$1$')) -Property access)){ $acl | Set-Acl -Path $remote}}
```

```
1..9|%{"{0,9} x 8 + $_ = $(8*($a=1..$_-join'')+$_)"-f$a}
```

TIPS AND TRICKS #1

- Prefix/Suffix commands with Get-Date to record when ran for cross referencing
  - Get-Date; Test-NetConnection dodgyserver
  - Or set in Prompt() function in profile ($profile)
- $PSVersionTable
  - See what PoSH version you are running
- Ctrl r to search persistent history
- Tab complete & find Windows commands as well as PoSH ones
- Measure-Object
- Measure-Command
- Out-Gridview (-PassThru)

TIPS AND TRICKS #2

- Ctrl Backspace/Delete to delete whole word back/forward
- Ctrl Home/End to delete to start/end of line
- Ctrl z/y – undo (multiple)/redo
- Ctrl arrow – jump words
- Number conversions
  - '{0:x}' –f 1234
  - 0x4d2
  - [convert]::ToString(1234 , 2)
- scb/gcb – pipe to/from clipboard
- gcb | ConvertFrom-JSON
- (gcb).length – how long is that string in the clipboard
- (get-date) - (gcim Win32_OperatingSystem).LastBootUpTime – what's the uptime?

RUNNING POWERSHELL SCHEDULED TASKS

- Use to pre-empt problems, alert, fix, tidy, etc
- Use a service account with required permissions & group memberships
- Use a single machine, install all required modules, snapins, etc
- Create your own scheduled tasks folder for your tasks
- Make the script write a log file (Start-Transcript is good enough)
- Avoid clear text credentials in command lines – Secure String/Protect-CmsMessage
- Ensure service account has "Log on as a batch job" privilege
- How do you detect if scheduled task has failed?
- Test command line in cmd.exe, e.g. powershell.exe –file
- Parameter gotchas
  - $true/$false being treated as strings not bools
  - Arrays (comma separated) flattened into a single element
  - Quotes/spaces (or lack there of)
- Use PowerShell to create the scheduled task?
- Ensure script doesn't prompt

OTHER USEFUL SCRIPTS

- Show registry keys modified in a given time period
- Show files changed since boot (e.g. Citrix PVS cache) or any time period
- Get chunk at file offset (e.g. tally procmon write to log file to contents)
- Digital clock/stopwatch/countdown timer (e.g. time & mark logon stages)
- Show/search loaded modules for processes (e.g. find hook dlls)
- Trim process working sets, set working set size limits (e.g. save memory)
- Get installed software (do not use Win32_Product) (e.g. compare systems)
- Change CPU priorities (e.g. stop impact of runaway processes)
- Find/delete user profiles (e.g. infrastructure servers short of free disk space)

HOW TO IMPROVE AT POWERSHELL

- Sit in PowerShell prompt not cmd
- Use it every working day
- Look at other people's scripts and understand them (how I started in 1980)
- Don't ignore automation opportunities but not everything is
- Understand and learn from errors – don't run away screaming
- Don't give up
- Bite the bullet – don't go back to the old ways of cmd, vbs, etc
- Everyone had to start somewhere
- Online training
- Communities – give and take (usually the other way round initially)
- Books

FOOD FOR THOUGHT

- PowerShell is cross-platform – Mac & Linux too
- PowerShell is open source - https://github.com/powershell
- PowerShell v5.1 (latest/last Windows release) is EoL
  - PowerShell 7 is here (pwsh)
- Cmd batch scripting is painful & needs lots of exes for troubleshooting
- Powershell.exe can be slow to start compared with wscript.exe & cmd.exe
- Easy to create HTML/CSV and send SMTP emails – be proactive!

RESOURCES

- https://www.controlup.com/script-library/
- https://github.com
- https://gallery.technet.microsoft.com/scriptcenter (deprecated)
- https://devblogs.microsoft.com/scripting/
- https://4sysops.com
- Your preferred search engine

EXIT(0)

- Thank you

- Live long and PowerShell