

Using PowerShell to Collect System Information



Liam Cleary

Microsoft MVP and Microsoft Certified Trainer at SharePlicity

@helloitsliam | www.helloitsliam.com



Overview

Goal: Collect Initial Triage Data

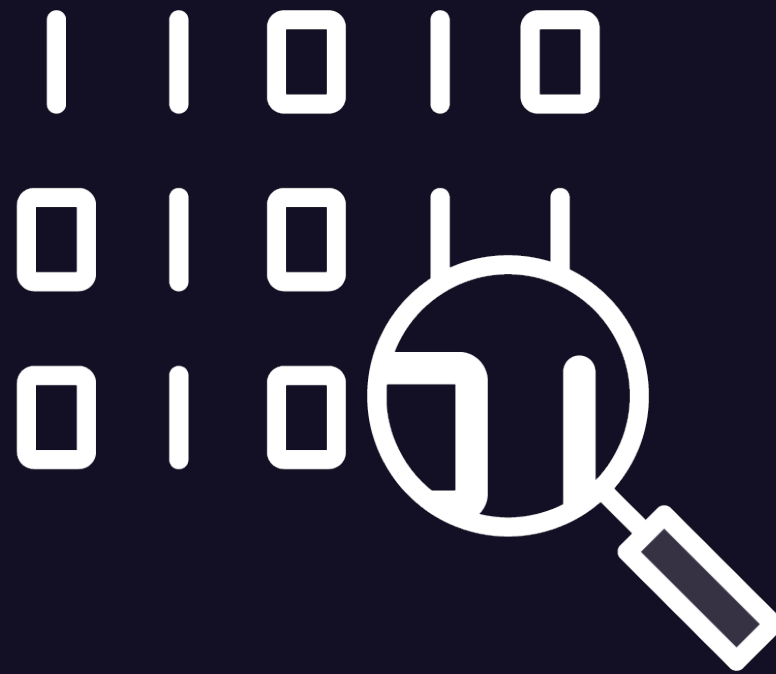
- Review required triage data
- Review available native PowerShell commands
- Execute PowerShell commands for system information retrieval
- Review supporting tools
- Understand how to use supporting tools with PowerShell
- Execute supporting tools
- How to format the retrieved information





Review Required Triage Data



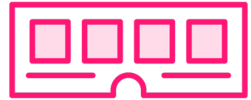


Digital or Forensic Triage

The process of collecting, assembling, analyzing, and prioritizing digital evidence for an investigation.



Triage Data Categories



Volatile

Pulled out of memory using system commands.
Temporary and usually disappears quickly.



Windows and File System

Collect core windows and file system information such as the Windows Registry, Prefetch files, and Event logs.



Persistence

Processes or tasks that run automatically.

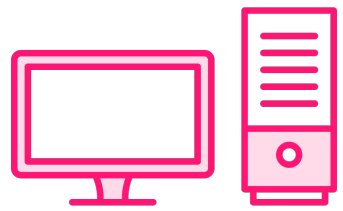


Application-specific

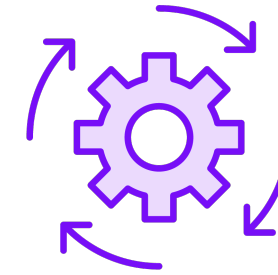
Installed or updated applications, browsing history, or application-specific configuration.



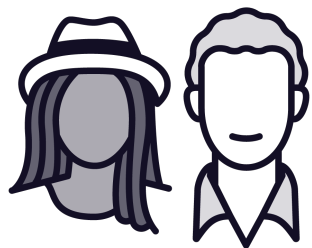
Required Information



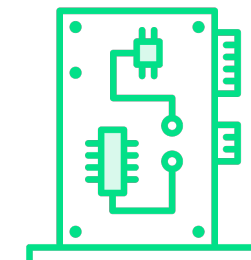
System Information



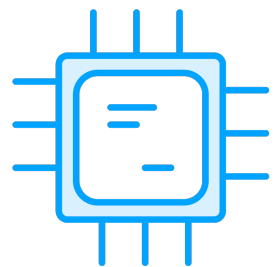
Processes and Services



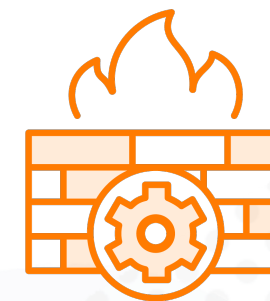
Local Users, Groups, and Memberships



Network Adapters and Connections



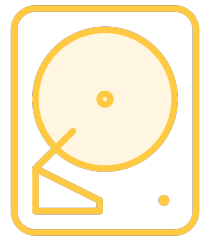
Processor, BIOS, and Memory Details



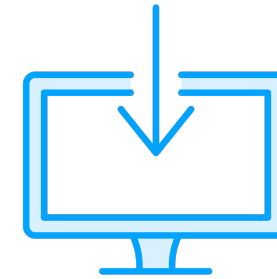
Firewall Configuration



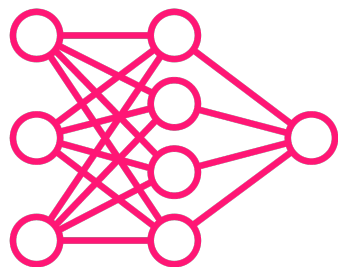
Required Information



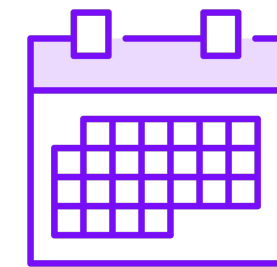
Hard Disk, Folders, and File Information



Installed Applications



Current Session Information



Tasks and Autorun Information



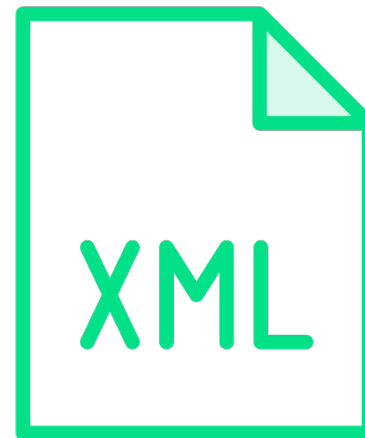
Group Policy Details



Event Logs



Output Options

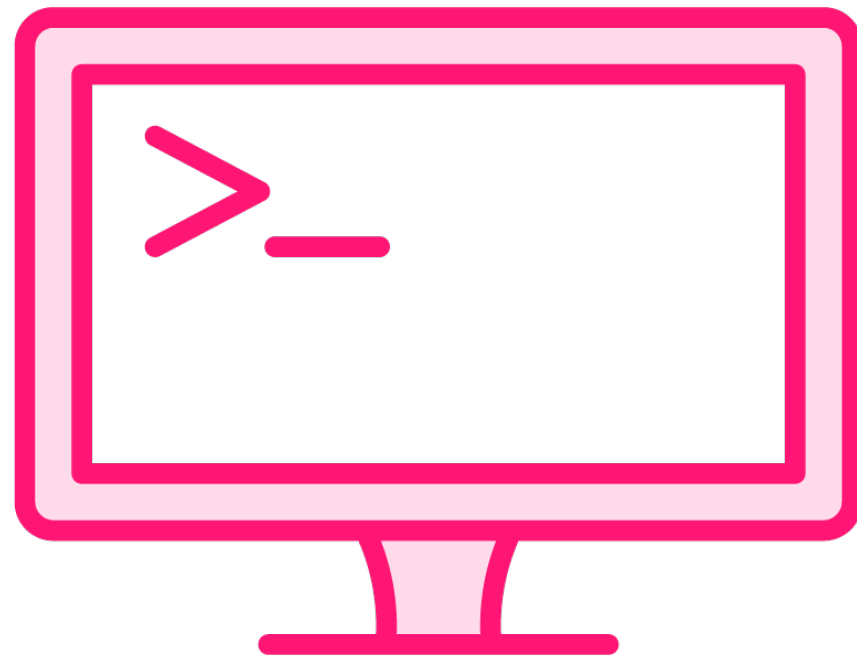




Review Available Native PowerShell Commands



Common PowerShell Commands



Retrieve local users and groups

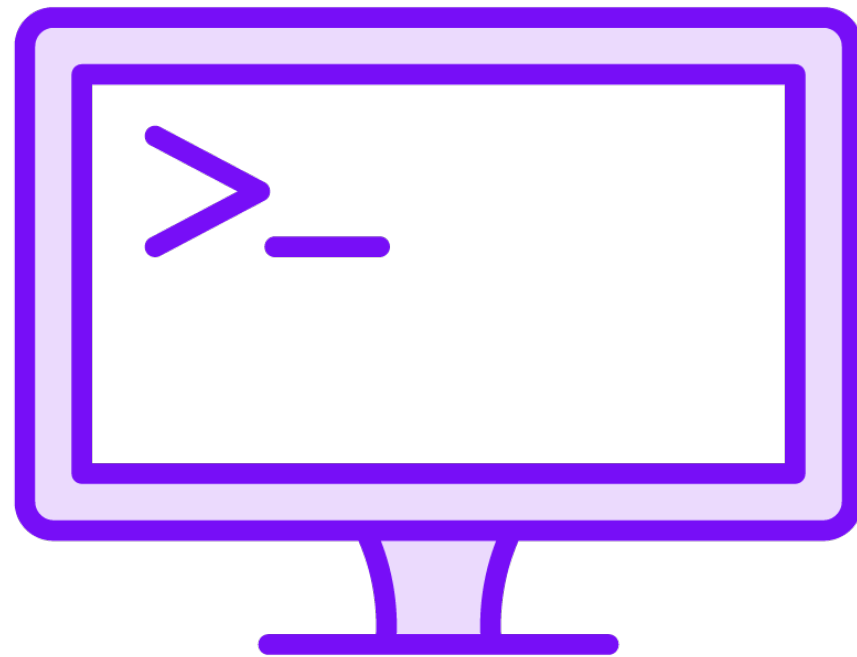
- Get-LocalUser
- Get-LocalGroup
- Get-LocalGroupMember

Workstation information

- Get-CimInstance Win32_OperatingSystem
- Get-CimInstance Win32_Processor
- Get-CimInstance Win32_BIOS
- Get-CimInstance Win32_LogicalDisk
- Get-CimInstance Win32_ComputerSystem



Common PowerShell Commands



Running services and processes

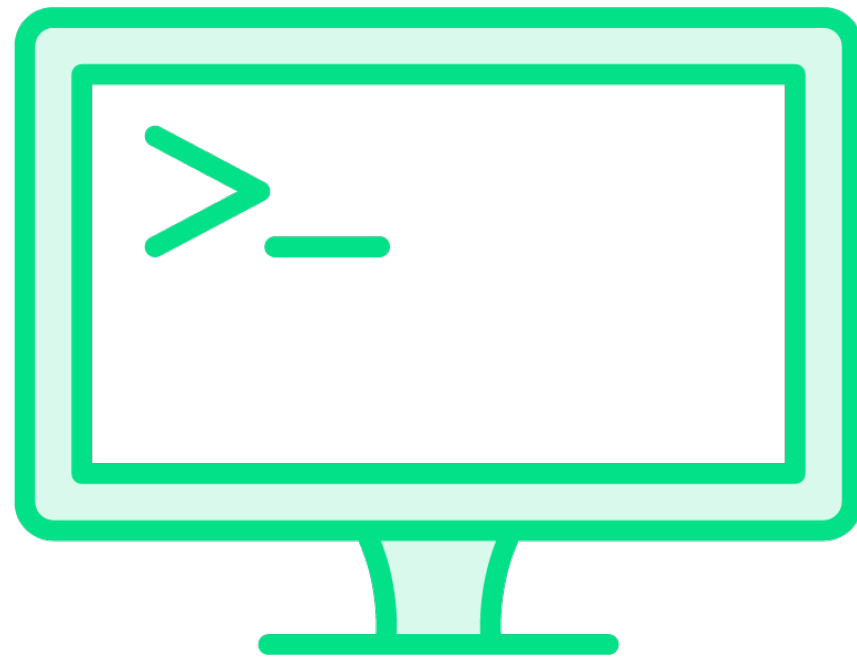
- Get-CimInstance Win32_Service
- Get-CimInstance Win32_Process
- Get-Process
- Get-Service

Networking information

- Get-CimInstance Win32_NetworkAdapter
- Get-NetIPAddress
- Get-DnsClientCache
- Get-NetNeighbor
- Get-NetRoute



Common PowerShell Commands



Scheduled tasks and autoruns

- `Get-CimInstance Win32_StartupCommand`
- `Get-CimInstance Win32_ScheduledJob`

Files, folders, and sessions

- `Get-ChildItem`
- `Get-SmbSession`
- `Get-CimInstance Win32_Share`

Event logs

- `Get-WinEvent`



Execute PowerShell Commands for System Information Retrieval



Using Native Commands

Retrieve first 10 running processes and service

```
Get-Process | Select-Object -First 10
```

```
Get-Service | Select-Object -First 10
```

Retrieve local users and groups

```
Get-LocalUser | Select-Object Name, Enabled | Format-Table -AutoSize
```

```
Get-LocalGroup | Select-Object Name, SID | Format-Table -AutoSize
```

Retrieve computer information

```
Get-ComputerInfo
```

Retrieve computer network information

```
Get-NetAdapter | Select-Object Name, Status, LinkSpeed
```

```
Get-NetIPAddress | Select-Object IPAddress, InterfaceIndex, AddressFamily
```

```
Get-NetRoute
```



Using CIM Instance Commands

Retrieve first 10 running processes and service

```
Get-CimInstance -ClassName Win32_Process | Select-Object -First 10
```

```
Get-CimInstance -ClassName Win32_Service | Select-Object -First 10
```

Retrieve local users and groups

```
Get-CimInstance Win32_UserAccount | Select-Object Name, Caption
```

```
Get-CimInstance Win32_Group | Select-Object Name, Caption
```

Retrieve computer information

```
Get-CimInstance -ClassName Win32_ComputerSystem
```

```
Get-CimInstance -ClassName Win32_Processor
```

```
Get-CimInstance -ClassName Win32_OperatingSystem
```



CIM Instance Command Versus Native Command

Cim Instance

Older commands

Common interface for retrieving information

Access to entire namespace of objects within computer

Support for more versions of Windows

VS

Native Command

Newer commands

Maybe modified after updating or patching

Limited to specific types of information

May not have a command to retrieve specific information



Demo

Execute PowerShell Commands for System Information Retrieval

- Using native Windows PowerShell commands
- Using CIM Instance PowerShell commands

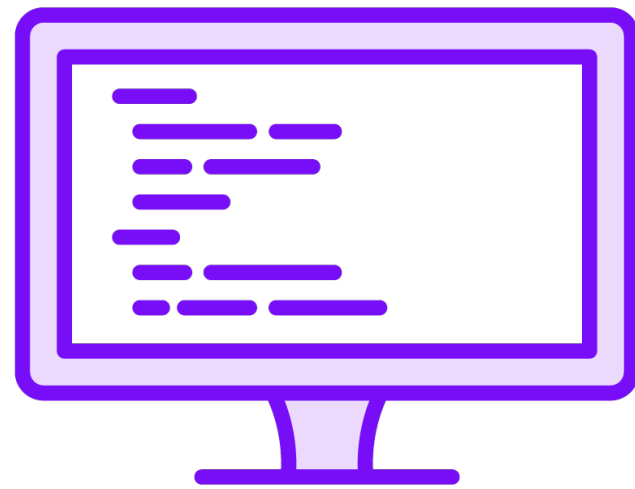




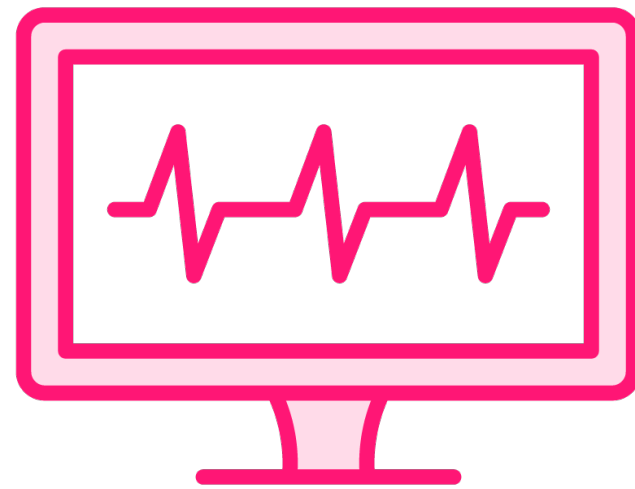
Review Supporting Tools



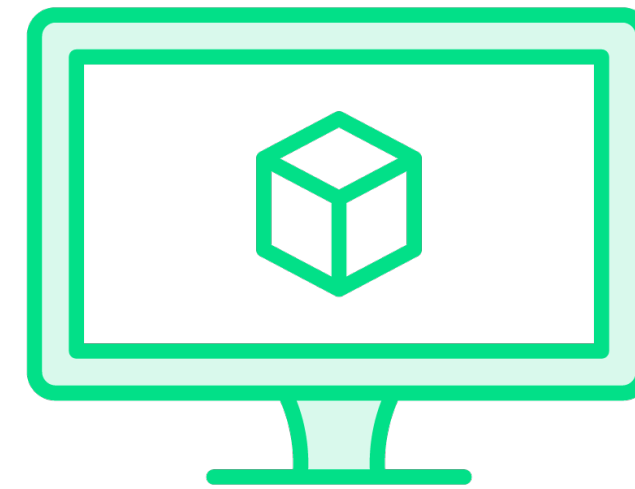
Supporting Tool Options



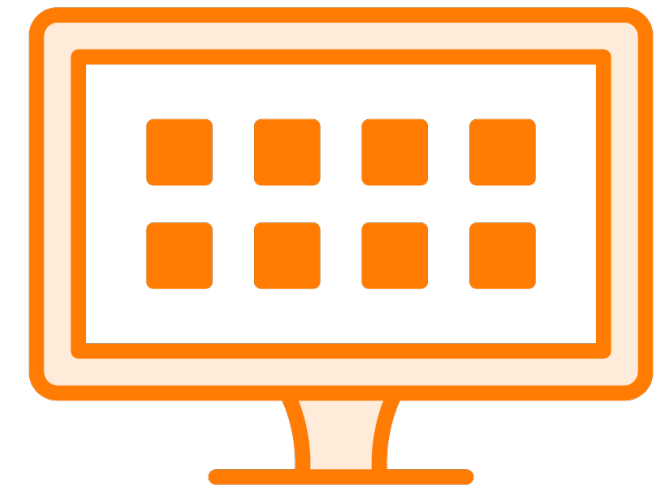
**Downloadable
Scripts**



**Microsoft
Sysinternals**

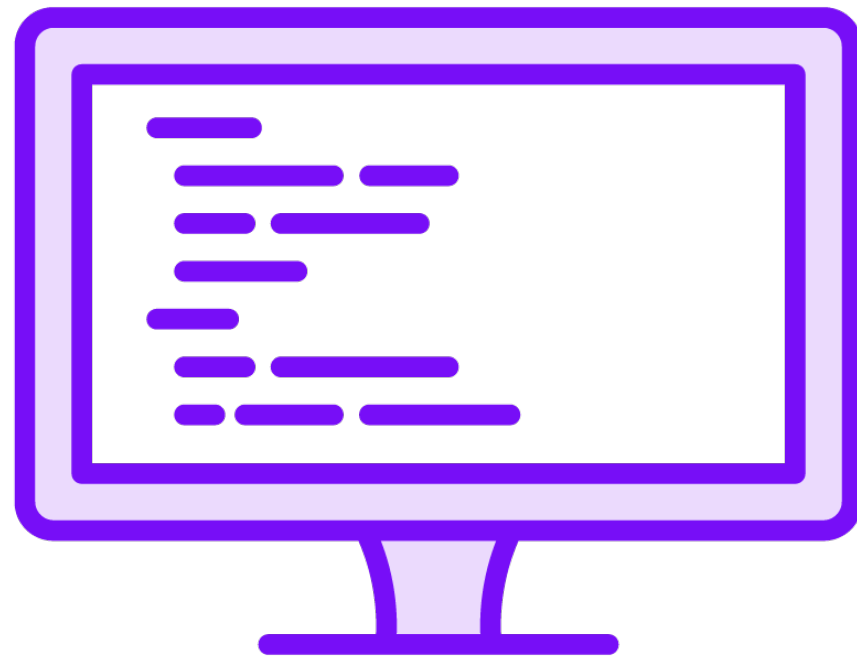


**Open-source
Applications**



**Vendor
Applications**

Downloadable Scripts



Many available in most free code repositories

Written from real-world experience

Some maintained, many not

Fills gaps in the basic PowerShell scripting process

Provides enhanced abilities for output

Need to review the scripts before execution



Microsoft Sysinternals

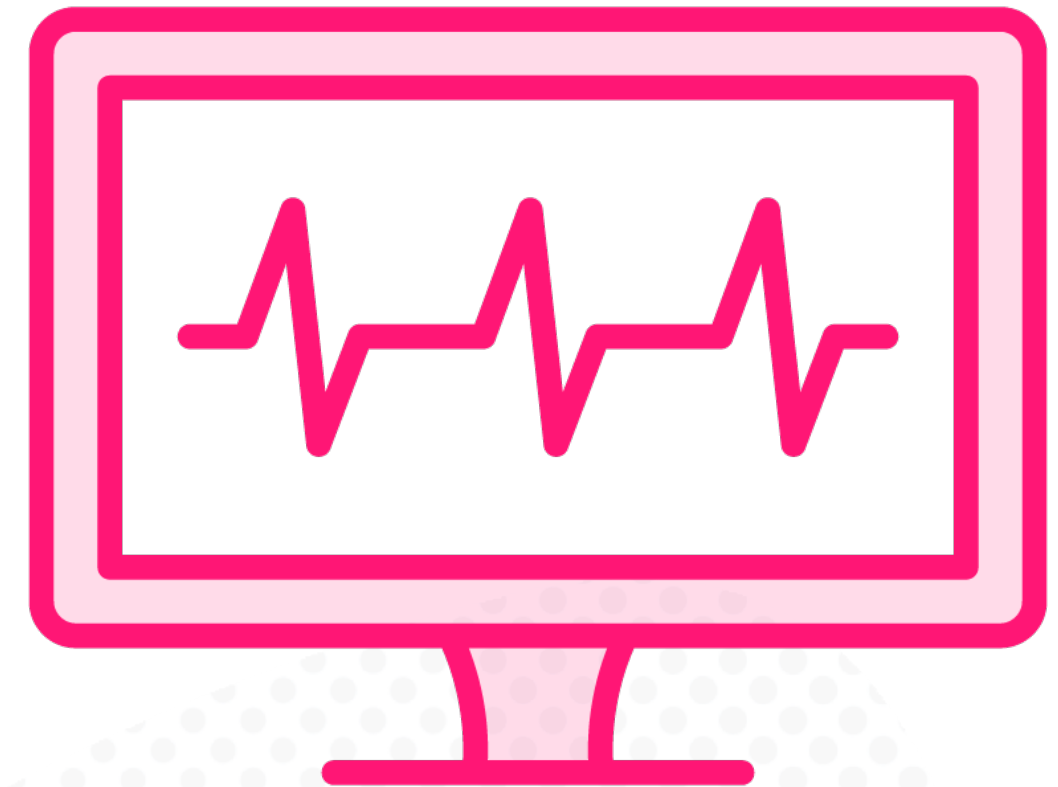
Freely available and maintained by Microsoft

Designed for support and troubleshooting

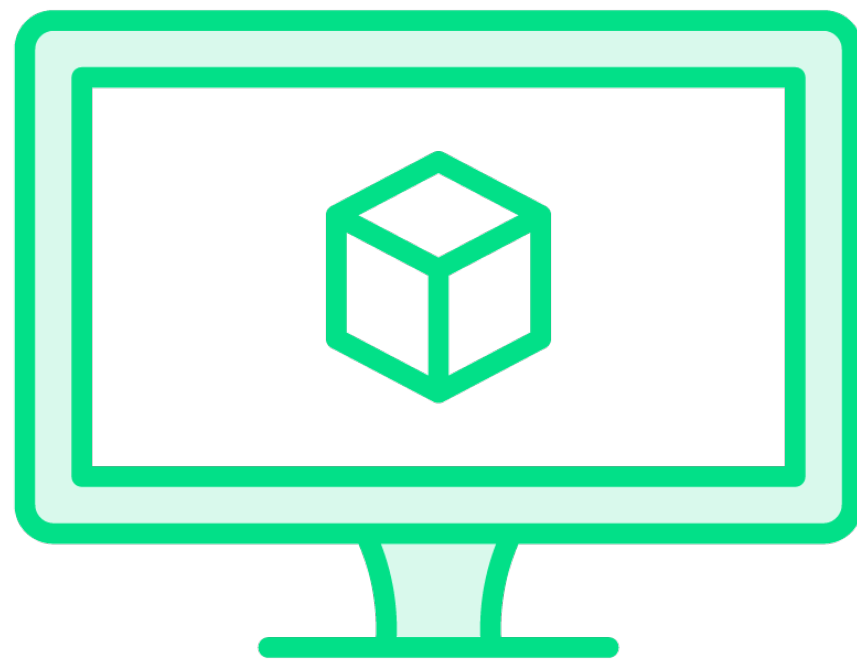
Not specifically forensic or triage
applications

Provides enhanced information inspection
and gathering

Performs tasks not available directly within
PowerShell



Open-source Applications



Multiple open-source applications available

Use web browser to search for 'open-source windows triage application'

Not all maintained

Contain most features required to triage a workstation

Provides case management



Vendor Applications

Specifically design for triage, and forensics

Built to support much more than just
information gathering

Provides case management

Supports the full investigation, triage, and
forensic workflow



Supporting Tools



Microsoft Sysinternals



PowerForensics

Understand How to Use Supporting Tools with PowerShell



Microsoft Sysinternals

Microsoft bundled the Sysinternals Utilities into a single download. The download contains troubleshooting utilities and help files only.

<https://download.sysinternals.com/files/SysinternalsSuite.zip>



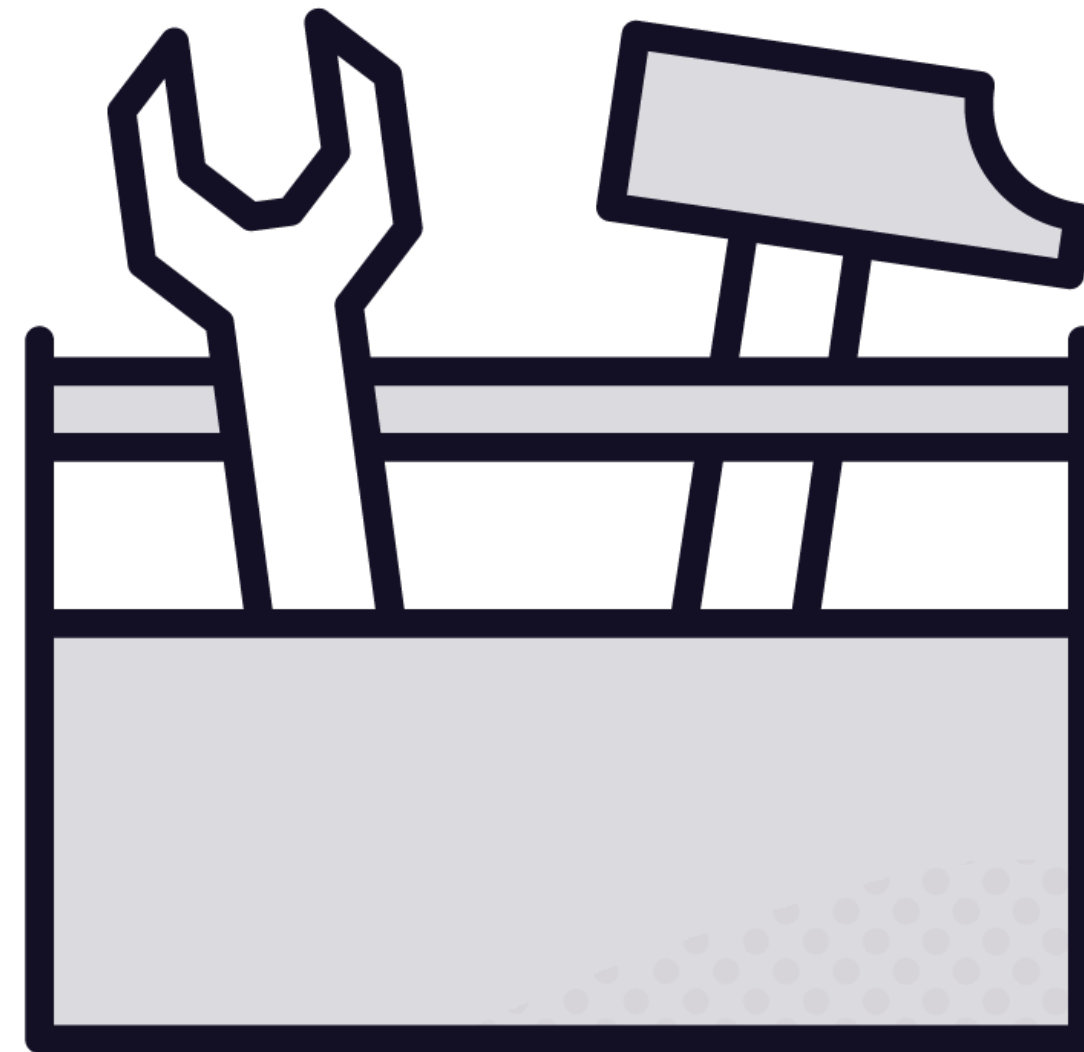
Download from Microsoft

Extract downloaded file

**Execute each command
either directly or within a
console**

**Initial execution requires
acceptance of the usage
banner**

**Commands will work
within 'cmd' and 'pwsh'**



Sysinternal Tools

- 1 **Support for command line execution**
- 2 **Supports various command line arguments**
- 3 **Support for exporting the results**



Example of Executing a Sysinternals Tool

Review auto running process

```
.\autorunsc.exe /accepteula
```

Review all auto running processes and hide Microsoft entries

```
.\autorunsc.exe -a * -m /accepteula
```

Review auto running process and select properties using PowerShell

```
.\autorunsc.exe -a * -m -c /accepteula | ConvertFrom-Csv | `
    Select-Object "Entry", "Description", "Image Path", "Enabled"
```





Execute Supporting Tools



Execute Supporting Tools



Sysinternals utility output includes banner



Not all utilities support command line execution



Don't all support CSV, XML, or HTML output



Properties are not always available for querying



Removing the Sysinternals Banner

```
LogonSessions v1.41 - Lists logon session information
Copyright (C) 2004-2020 Mark Russinovich
Sysinternals - www.sysinternals.com

Logon Session,User Name,Auth Package,Logon Type,Session,Sid,Logon
00000000:000003e7,WORKGROUP\WIN10$,NTLM,(none),0,S-1-5-18,1/20/2023
00000000:0000d7e3,,NTLM,(none),0,(none),1/20/2023 8:31:24 AM,,,,
00000000:0000dc09,Font Driver Host\UMFD-1,Negotiate,Interactive,1,
00000000:0000dc0a,Font Driver Host\UMFD-0,Negotiate,Interactive,0,
00000000:000003e4,WORKGROUP\WIN10$,Negotiate,Service,0,S-1-5-20,1,
00000000:0001396d,Window Manager\DWM-1,Negotiate,Interactive,1,S-1
00000000:00013985,Window Manager\DWM-1,Negotiate,Interactive,1,S-1
00000000:000003e5,NT AUTHORITY\LOCAL SERVICE,Negotiate,Service,0,S
00000000:0006fb59,NT Service\MSSQL$SQLEXPRESS,Negotiate,Service,0,
31323133,1/20/2023 8:33:28 AM,,,,
00000000:0007b99b,NT Service\MSSQLFDLauncher$SQLEXPRESS,Negotiate,
446180-2226563786,1/20/2023 8:33:29 AM,,,,
00000000:0007edda,NT Service\SQLTELEMETRY$SQLEXPRESS,Negotiate,Se
398-3434236965,1/20/2023 8:33:30 AM,,,,
00000000:0009595d,NT Service\MSSQLLaunchpad$SQLEXPRESS,Negotiate,S
914861-206046543,1/20/2023 8:33:32 AM,,,,
00000000:02328e9b,WIN10\Trainer,NTLM,Interactive,1,S-1-5-21-411590
N10,,,
00000000:02328ec6,WIN10\Trainer,NTLM,Interactive,1,S-1-5-21-411590
N10,,,
```



```
Logon Session,User Name,Auth Package,Logon Type,Session,Sid,Logon
00000000:000003e7,WORKGROUP\WIN10$,NTLM,(none),0,S-1-5-18,1/20/2023
00000000:0000d7e3,,NTLM,(none),0,(none),1/20/2023 8:31:24 AM,,,,
00000000:0000dc09,Font Driver Host\UMFD-1,Negotiate,Interactive,1,
00000000:0000dc0a,Font Driver Host\UMFD-0,Negotiate,Interactive,0,
00000000:000003e4,WORKGROUP\WIN10$,Negotiate,Service,0,S-1-5-20,1,
00000000:0001396d,Window Manager\DWM-1,Negotiate,Interactive,1,S-1
00000000:00013985,Window Manager\DWM-1,Negotiate,Interactive,1,S-1
00000000:000003e5,NT AUTHORITY\LOCAL SERVICE,Negotiate,Service,0,S
00000000:0006fb59,NT Service\MSSQL$SQLEXPRESS,Negotiate,Service,0,
31323133,1/20/2023 8:33:28 AM,,,,
00000000:0007b99b,NT Service\MSSQLFDLauncher$SQLEXPRESS,Negotiate,
446180-2226563786,1/20/2023 8:33:29 AM,,,,
00000000:0007edda,NT Service\SQLTELEMETRY$SQLEXPRESS,Negotiate,Se
398-3434236965,1/20/2023 8:33:30 AM,,,,
00000000:0009595d,NT Service\MSSQLLaunchpad$SQLEXPRESS,Negotiate,S
914861-206046543,1/20/2023 8:33:32 AM,,,,
00000000:02328e9b,WIN10\Trainer,NTLM,Interactive,1,S-1-5-21-411590
N10,,,
00000000:02328ec6,WIN10\Trainer,NTLM,Interactive,1,S-1-5-21-411590
N10,,,
```

Utilize the -nobanner property



Execute Supporting Tools and Export

Retrieve TCP Port information

```
.\tcpvcon64.exe -a -c /accepteula | `
    Out-File -FilePath .\PortInformation.log
```

Retrieve open files by process

```
.\handle.exe -v /accepteula | `
    Out-File -FilePath .\OpenProcesses.log
```

Retrieve event log list

```
wmic nteventlog list brief | `
    Out-File -FilePath .\EventLogList.log
```



Demo

Execute Supporting Tools

- tcpvcon64.exe
- autorunsc.exe
- psloggedon64.exe
- logonsessions64.exe
- handle.exe
- klist.exe
- wmic

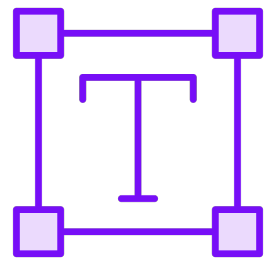




How to Format the Retrieved Information



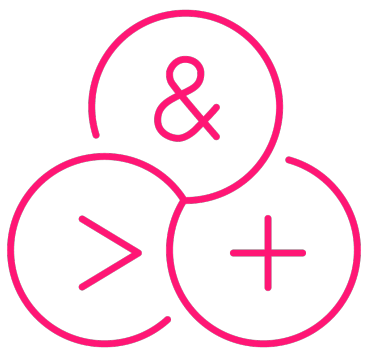
Formatting Data



Convert to string value using 'Out-String'

{JSON}

Convert to CSV, XML, HTML, or JSON using either ConvertFrom-Csv, ConvertTo-Xml, ConvertTo-Html, and ConvertTo-Json



Custom PowerShell function to convert to any type



Converting PowerShell Output

Retrieve local users and convert to HTML

```
Get-LocalUser | Select-Object -Property Name, Enabled | ConvertTo-HTML
```

Retrieve security event log and convert to CSV

```
Get-WinEvent -FilterHashtable @{ Logname = 'Security'; } | ConvertTo-Csv
```

Retrieve running services and convert to XML

```
Get-Service | ConvertTo-Xml -As String
```



Converting Sysinternals Tool Output

Retrieve TCP Port information and convert to Csv

```
.\tcpvcon64.exe -a -c /accepteula -nobanner | Out-String | `
    ConvertFrom-Csv -Delimiter ',' `
    -Header Protocol, Process, ID, State, Local, Remote
```

Retrieve event log list

```
wmic nteventlog list brief /format:csv | `
    Where-Object {$_. -ne ''} | `
    ConvertFrom-Csv | `
    Select-Object LogFileName, Name, NumberOfRecords, FileSize | `
    ConvertTo-Html -Fragment
```



Demo

Format the Retrieved Information

- Format to XML
- Format to CSV
- Format to HTML



Summary

Goal: Collect Initial Triage Data

- Reviewed the required triage data
- Executed standard PowerShell commands to retrieve workstation details
- Executed specific Sysinternals tools for more detailed system information
- Formatted the retrieved data in different types



Up Next:

Creating a Triage Script to Collect System Information

