Advanced Scripting   
Events and Logs

Last Updated: 4/18/2024 9:55 AM Version 1  
Document Prepared for: CYBER360 Student

# Name Click here to enter name ID Click here to enter id

# Instructions

Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

Microsoft Windows uses structured data objects to audit records and log system events. Linux/Unix syslogs use text strings for event logging. Let’s explore.

# Requirements

Windows  
Linux or macOS

# Task 1—Windows Event Viewer

To find the Event Viewer app, tap the [Flag] key and start typing “Event Viewer.” (Open it when it appears under Best Match.) It will look something like this:

A screenshot of a computer

Description automatically generated

## Steps

1. In the left pane under Event Viewer (Local), tap to expand “Windows Logs,” then tap System. The system events will appear in the middle pane. By default, they are sorted in reverse-chronological order.
   1. What is the Event ID of the most recent of your System logs? Click or tap here to enter text.
2. In the right pane under Actions, tap Find, then in the Find what: dialog type WindowsUpdate (no space between the words). Then tap [Find Next], then [Cancel].
   1. What is the Event ID of the most recent Windows Update event in your System logs? Click or tap here to enter text.
   2. Look at the detailed event view at the bottom of the middle pane. Next to the General tab, tap the Details tab, then select the “XML View” radio button. What is the value of the xmlns attribute of the root-level “Event” tag? Click or tap here to enter text.
   3. Tap the General tab to go back to the default event browsing display.
3. Take a moment to explore other Windows events on your system.
   1. Under Windows Logs, see what kinds of Application, Security, and Setup events you can find.
   2. Under Applications and Services Logs, see what kinds of logs you notice from various hardware and software vendors. In particular, see how many different Event ID numbers are being collected for *Windows PowerShell* and *PowerShellCore/Operational*.

# Task 2—Legacy CLI Windows Event Queries with **wevtutil**

In Windows, launch two command-line sessions: one without elevated privilege, and another with Administrator privilege. For this task, it doesn’t matter whether each command-line session is legacy CMD.EXE or PowerShell (core or desktop), but most students understandably prefer PowerShell.

## Steps

1. To see wevtutil’s usage help, execute it by itself or with its /? switch argument:  
   **wevtutil**   
   **wevtutil /?**
2. Get the configuration (in XML format) for the System and Application events:   
   **wevtutil get-log System /format:xml**   
   **wevtutil gl Application /f:xml**
   1. What is the filename extension of each event log file, as shown in the text of the <logFileName> XML tag?
3. In the unprivileged CLI session, try getting the configuration for the Security events:   
   **wevtutil gl Security**
   1. Try again in the Administrator privileged CLI session. Did it work? Click or tap here to enter text.
4. Get the six newest events from Microsoft-Windows-PowerShell/Operational. Enter:   
   **wevtutil qe /rd:true /c:6 /f:text Microsoft-Windows-PowerShell/Operational**
   1. Try it again without the **/rd:true** option to see the six *oldest* PowerShell Operational events.
   2. Try it again for PowerShell Core Operational events:   
      w**evtutil qe /rd:true /c:6 /f:text PowerShellCore/Operational**
5. Close both CLI sessions (unprivileged and Administrator).

# Task 3—PowerShell Event Queries

In Windows, launch an Administrative-privileged PowerShell (core or desktop) session.

## Steps

1. Get a list of available logs:   
   **Get-WinEvent -ListLog \* 2>C:\TEMP\errors.txt**
   1. Look in the contents of the file **C:\TEMP\errors.txt**. What logs were you unable to probe, even with Administrator privilege? Click or tap here to enter text.
   2. Count the event logs:   
      **Get-WinEvent -ListLog \* | measure**   
      How many? Click or tap here to enter text.
   3. Sort them by number of events:   
      **Get-WinEvent -Listlog \* | sort RecordCount**   
      On *your* system, which event file has the most events? Click or tap here to enter text.
2. How many PowerShell logs are there, and what are their names? Let’s use the Where-Object cmdlet to filter for specific log names:   
   **Get-WinEvent -ListLog \*| ? {$\_.LogName -match "PowerShell"}**   
   (*Reminder* **?** *is one of the convenient aliases for* ***Where-Object***.)
   1. How many are there? Click or tap here to enter text.
   2. What are their names (joined on one line, comma separated)? Click or tap here to enter text.
3. Get the three newest events from Microsoft-Windows-PowerShell/Operational. Enter:   
   **Get-WinEvent Microsoft-Windows-PowerShell/Operational -MaxEvents 3**
   1. Try it again with the **-Oldest** switch parameter to see the three *oldest* PowerShell Operational events.
   2. Write a **Get-WinEvent** command line to find the three newest events from PowerShellCore/Operational. Your command: Click or tap here to enter text.

# Task 4—Interesting Windows Event IDs

In Windows, launch an Administrative-privileged PowerShell (core or desktop) session.

For each of the following, browse some of the events in the named log file, and see if you can find any logs with the given Event ID. For any that you successfully find, try to interpret what the event means, and share your interpretation in just one or two sentences. If you have difficulty trying to interpret an event, get some help from classmates, Internet searches, and generative-AI chatbots. (For any Event IDs that you cannot find, just fill in “NA” or “not applicable” as your interpretation.)

## Steps

1. “Windows PowerShell”
   1. Did you find any events with Event ID 200? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   2. Event ID 400? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   3. Event ID 800? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
2. Microsoft-Windows-PowerShell/Operational *and* PowerShellCore/Operational
   1. Event ID 4103? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   2. Event ID 4104? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   3. Event ID 4105? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   4. Event ID 4106? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   5. Event ID 40961? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   6. Event ID 40962? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
3. Security
   1. Event ID 4657? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   2. Event ID 4688? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   3. Event ID 1100? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   4. Event ID 1102? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   5. Event ID 1104? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   6. Event ID 4624? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   7. Event ID 4625? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
4. System
   1. Event ID 104? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   2. Event ID 5719? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
5. “Microsoft-Windows-Windows Defender/Operational”
   1. Event ID 1116? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   2. Event ID 1117? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.
   3. Event ID 5007? Click or tap here to enter text.   
      Your interpretation: Click or tap here to enter text.

# Task 5—Syslog, Journald

In Linux and macOS, logged events are handled by *syslog* or one of its variants (syslog-ng, rsyslog, etc.). Many systemd-based Linux distributions use its *journald* service instead of (or in addition to) syslog.

Windows Subsystem for Linux (WSL) is just a subsystem, not a full-blown Linux system, so it might not enable a syslog service by default.

Use **pwsh** to launch PowerShell Core in your Linux or macOS environment.

## Steps

1. Almost all logs get stored in **/var/log** and its subfolders. Go and see what’s there:   
   **Set-Location /var/log; Get-ChildItem** # or try **cd /var/log; ls**
2. How you explore your logs will differ depending on your environment.
   1. If you are using WSL with Ubuntu, enter:   
      **Get-Content dpkg.log | Select-String tree** # or **cat dpkg.log | grep tree**   
      You should see the event strings that were logged when you installed the tree package during one of your week 1 exercises. (If you installed some other WSL distribution, see what you can find in other log files.)
   2. If you are using macOS or a complete Linux distribution (not WSL), try one of these:   
      **cat syslog** # common in macOS, and in Debian-style distributions.   
      **cat messages** # common in Red Hat-style distributions such as Fedora.   
      **journalctl** # common in systemd-based Linux distributions.   
      Filter the log file for a particular keyword. For example, to filter for systemd:   
      **gc syslog | Select-String systemd**   
      **gc messages | Select-String systemd**   
      **journalctl | Select-String systemd**

# Deliverable

Upload this document with completed answers to I-Learn Canvas.