Introduction to Log File Analysis

By Zach Mewshaw

Objectives:

- Understand what log files are on Windows and Linux operating systems.
- Understand the basics of analyzing log files in Linux command line and PowerShell.
- Understand the basics of scripting in Bash to aid in our analysis.

Introduction

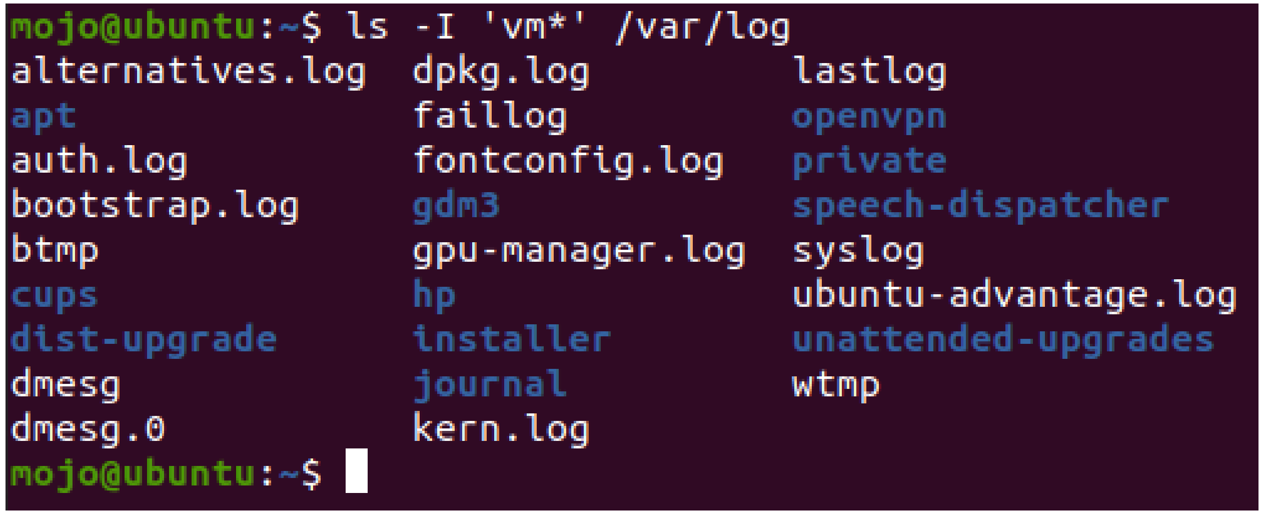
- “A log file is a file that records either events that occur in an operating system or other software runs, or messages between different users of a communication software.” (Wikipedia).

- Logs vs network traffic

- Used by:
  - **Cybersecurity Analysts**
  - **System Administrators**
  - **Penetration Testers**
  - **The list goes on**

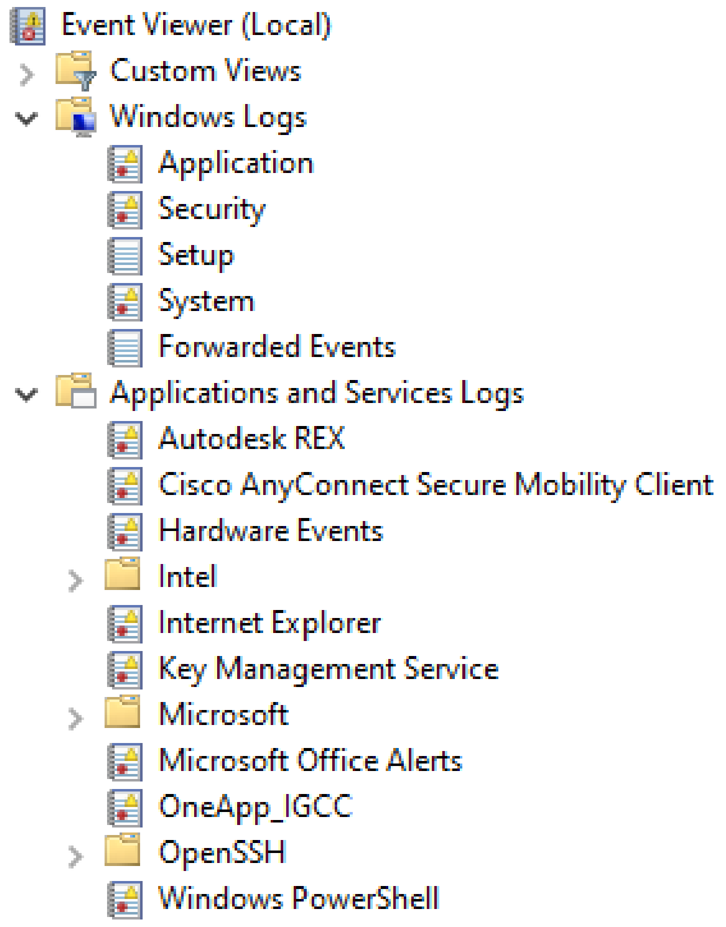
Introduction

Linux: /var/log



```
mojo@ubuntu:~$ ls -I 'vm*' /var/log
alternatives.log  dpkg.log          lastlog
apt               faillog           openvpn
auth.log          fontconfig.log    private
bootstrap.log     gdm3              speech-dispatcher
btmp              gpu-manager.log   syslog
cups              hp                ubuntu-advantage.log
dist-upgrade      installer         unattended-upgrades
dmesg             journal           wtmp
dmesg.0           kern.log
mojo@ubuntu:~$ 
```

Windows: Event Viewer



```
Event Viewer (Local)
> Custom Views
v Windows Logs
    Application
    Security
    Setup
    System
    Forwarded Events
v Applications and Services Logs
    Autodesk REX
    Cisco AnyConnect Secure Mobility Client
    Hardware Events
  > Intel
    Internet Explorer
    Key Management Service
  > Microsoft
    Microsoft Office Alerts
    OneApp_IGCC
  > OpenSSH
    Windows PowerShell
```

Linux

- Linux stores all log files in the directory /var/log.
- Seen far more often in CTF’s than Windows logs as most servers run on Linux.
- Important files:
  - **auth.log – Authorization and security related events**
  - **kern.log – Kernel messages**
  - **syslog – Everything**

Linux

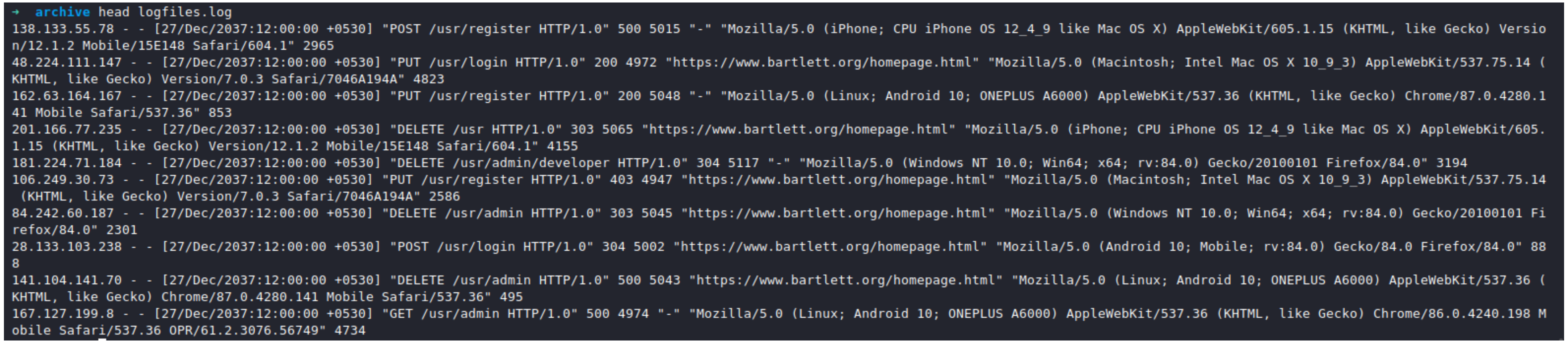
- We can analyze log files in Linux directly in the command line with text processing commands:
  - **cat – Print the file**
  - **awk – Show only the nth column in a text file**
  - **cut – Remove sections**
  - **sort – Sort lines**
  - **uniq – Omit repeated lines**
  - **wc – Print counts**
  - **less – Displays text in pages**

  grep – Print lines that match a pattern

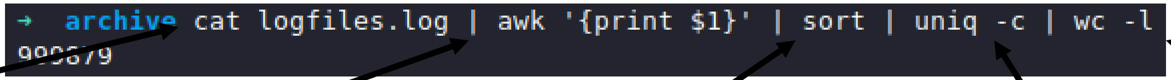
  tr – Translate or delete characters
- Piping allows us to chain these commands together!

Linux

Example randomly generated Apache web server log



```
→ archive head logfiles.log
138.133.55.78 - - [27/Dec/2037:12:00:00 +0530] "POST /usr/register HTTP/1.0" 500 5015 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 12_4_9 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Versio
n/12.1.2 Mobile/15E148 Safari/604.1" 2965
48.224.111.147 - - [27/Dec/2037:12:00:00 +0530] "PUT /usr/login HTTP/1.0" 200 4972 "https://www.bartlett.org/homepage.html" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.75.14 (
KHTML, like Gecko) Version/7.0.3 Safari/7046A194A" 4823
162.63.164.167 - - [27/Dec/2037:12:00:00 +0530] "PUT /usr/register HTTP/1.0" 200 5048 "-" "Mozilla/5.0 (Linux; Android 10; ONEPLUS A6000) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.1
41 Mobile Safari/537.36" 853
201.166.77.235 - - [27/Dec/2037:12:00:00 +0530] "DELETE /usr HTTP/1.0" 303 5065 "https://www.bartlett.org/homepage.html" "Mozilla/5.0 (iPhone; CPU iPhone OS 12_4_9 like Mac OS X) AppleWebKit/605.
1.15 (KHTML, like Gecko) Version/12.1.2 Mobile/15E148 Safari/604.1" 4155
181.224.71.184 - - [27/Dec/2037:12:00:00 +0530] "DELETE /usr/admin/developer HTTP/1.0" 304 5117 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0" 3194
106.249.30.73 - - [27/Dec/2037:12:00:00 +0530] "PUT /usr/register HTTP/1.0" 403 4947 "https://www.bartlett.org/homepage.html" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.75.14
 (KHTML, like Gecko) Version/7.0.3 Safari/7046A194A" 2586
84.242.60.187 - - [27/Dec/2037:12:00:00 +0530] "DELETE /usr/admin HTTP/1.0" 303 5045 "https://www.bartlett.org/homepage.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Fi
refox/84.0" 2301
28.133.103.238 - - [27/Dec/2037:12:00:00 +0530] "POST /usr/login HTTP/1.0" 304 5002 "https://www.bartlett.org/homepage.html" "Mozilla/5.0 (Android 10; Mobile; rv:84.0) Gecko/84.0 Firefox/84.0" 88
8
141.104.141.70 - - [27/Dec/2037:12:00:00 +0530] "DELETE /usr/admin HTTP/1.0" 500 5043 "https://www.bartlett.org/homepage.html" "Mozilla/5.0 (Linux; Android 10; ONEPLUS A6000) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/87.0.4280.141 Mobile Safari/537.36" 495
167.127.199.8 - - [27/Dec/2037:12:00:00 +0530] "GET /usr/admin HTTP/1.0" 500 4974 "-" "Mozilla/5.0 (Linux; Android 10; ONEPLUS A6000) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 M
obile Safari/537.36 OPR/61.2.3076.56749" 4734
```

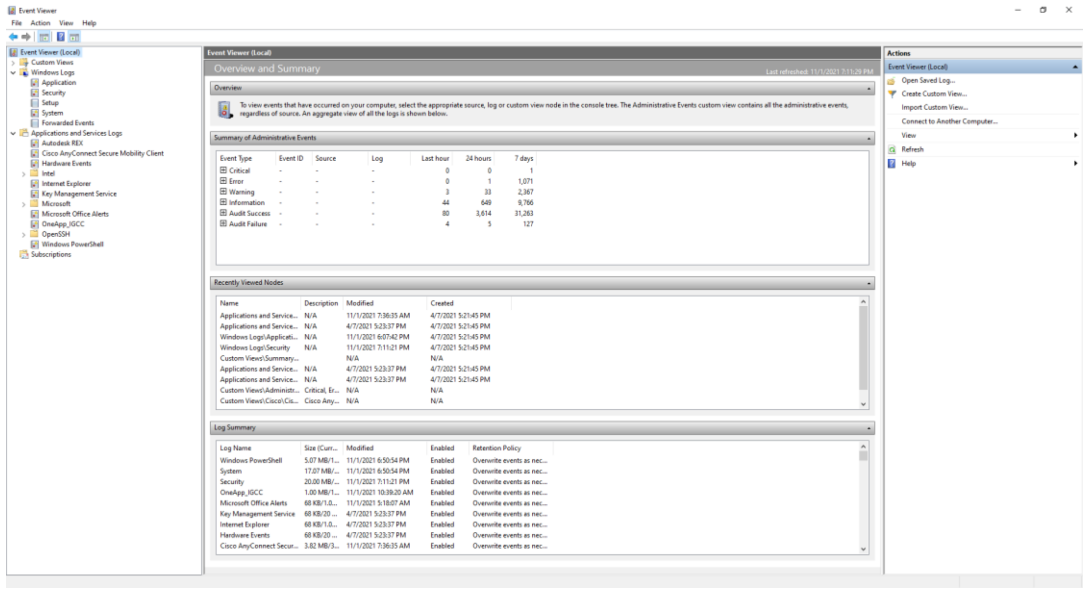


```
→ archive cat logfiles.log | awk '{print $1}' | sort | uniq -c | wc -l
999879
```

Output the file | **Show only the first column** | **Sort the data** | **Remove unique entries** | **Count the lines**

Windows Event Viewer (Vista+)

- Location: **C:\Windows\System32\winevt\Logs**
- Extension: **.evtx**
- Categories:
  - **Windows Logs**
  - **Applications and Services Logs**
  - **Extras**



Event Viewer (Local)

Overview and Summary

Last refreshed: 11/1/2021 7:11:29 PM

Overview

To view events that have occurred on your computer, select the appropriate source, log or custom view node in the console tree. The Administrative Events custom view contains all the administrative events, regardless of source. An aggregate view of all the logs is shown below.

Summary of Administrative Events

Event Type	Event ID	Source	Log	Last hour	24 hours	7 days
⊞ Critical	-	-	-	0	0	1
⊞ Error	-	-	-	0	1	1,071
⊞ Warning	-	-	-	3	33	2,367
⊞ Information	-	-	-	44	649	9,766
⊞ Audit Success	-	-	-	80	3,614	31,263
⊞ Audit Failure	-	-	-	4	5	127

Recently Viewed Nodes

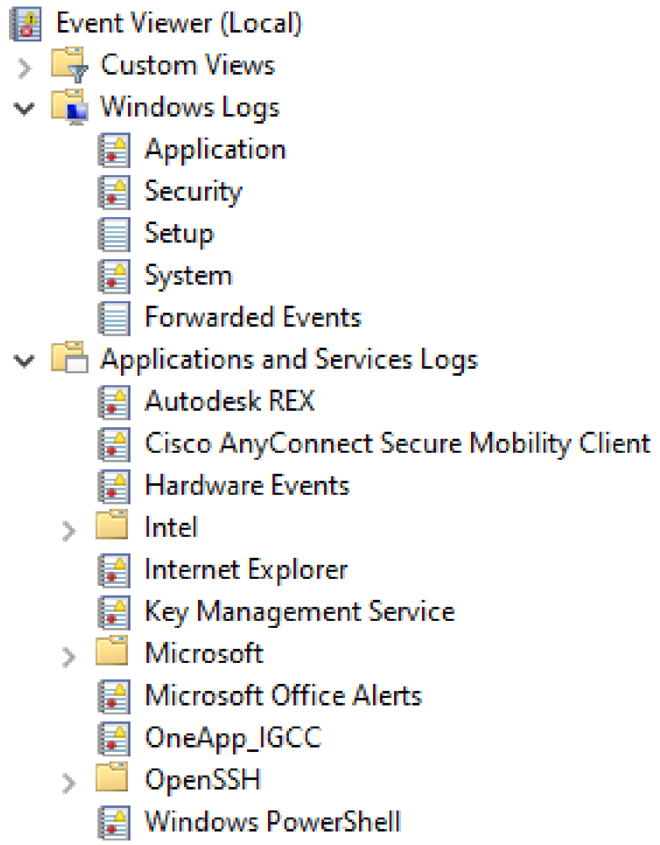
Name	Description	Modified	Created
Applications and Service...	N/A	11/1/2021 7:36:35 AM	4/7/2021 5:21:45 PM
Applications and Service...	N/A	4/7/2021 5:23:37 PM	4/7/2021 5:21:45 PM
Windows Logs\Applicati...	N/A	11/1/2021 6:07:42 PM	4/7/2021 5:21:45 PM
Windows Logs\Security	N/A	11/1/2021 7:11:21 PM	4/7/2021 5:21:45 PM
Custom Views\Summary...		N/A	N/A
Applications and Service...	N/A	4/7/2021 5:23:37 PM	4/7/2021 5:21:45 PM
Applications and Service...	N/A	4/7/2021 5:23:37 PM	4/7/2021 5:21:45 PM
Custom Views\Administr...	Critical, Er...	N/A	N/A
Custom Views\Cisco\Cis...	Cisco Any...	N/A	N/A

Log Summary

Log Name	Size (Curr...	Modified	Enabled	Retention Policy
Windows PowerShell	5.07 MB/1...	11/1/2021 6:50:54 PM	Enabled	Overwrite events as nec...
System	17.07 MB/...	11/1/2021 6:50:54 PM	Enabled	Overwrite events as nec...
Security	20.00 MB/...	11/1/2021 7:11:21 PM	Enabled	Overwrite events as nec...
OneApp_IGCC	1.00 MB/1...	11/1/2021 10:39:20 AM	Enabled	Overwrite events as nec...
Microsoft Office Alerts	68 KB/1.0...	11/1/2021 5:18:07 AM	Enabled	Overwrite events as nec...
Key Management Service	68 KB/20 ...	4/7/2021 5:23:37 PM	Enabled	Overwrite events as nec...
Internet Explorer	68 KB/1.0...	4/7/2021 5:23:37 PM	Enabled	Overwrite events as nec...
Hardware Events	68 KB/20 ...	4/7/2021 5:23:37 PM	Enabled	Overwrite events as nec...
Cisco AnyConnect Secur...	3.82 MB/3...	11/1/2021 7:36:35 AM	Enabled	Overwrite events as nec...

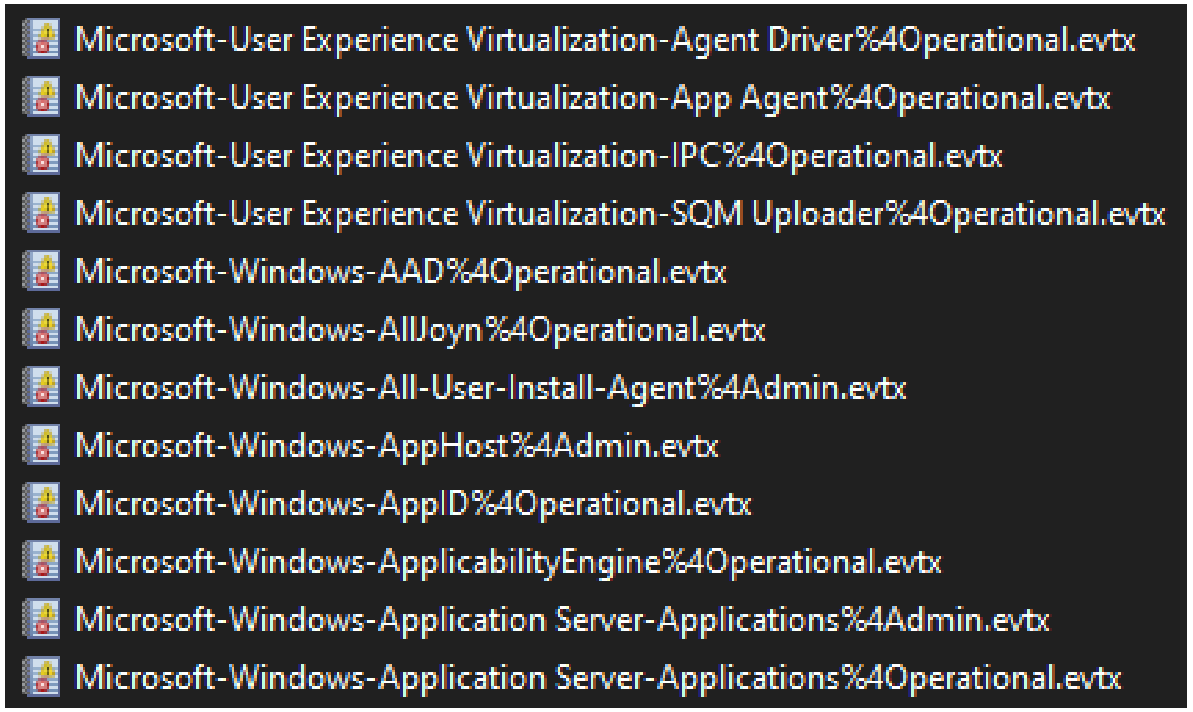
Windows Event Viewer (Vista+)

Event Viewer



```
Event Viewer (Local)
  > Custom Views
  v Windows Logs
        Application
        Security
        Setup
        System
        Forwarded Events
  v Applications and Services Logs
        Autodesk REX
        Cisco AnyConnect Secure Mobility Client
        Hardware Events
      > Intel
        Internet Explorer
        Key Management Service
      > Microsoft
        Microsoft Office Alerts
        OneApp_IGCC
      > OpenSSH
        Windows PowerShell
```

File Explorer

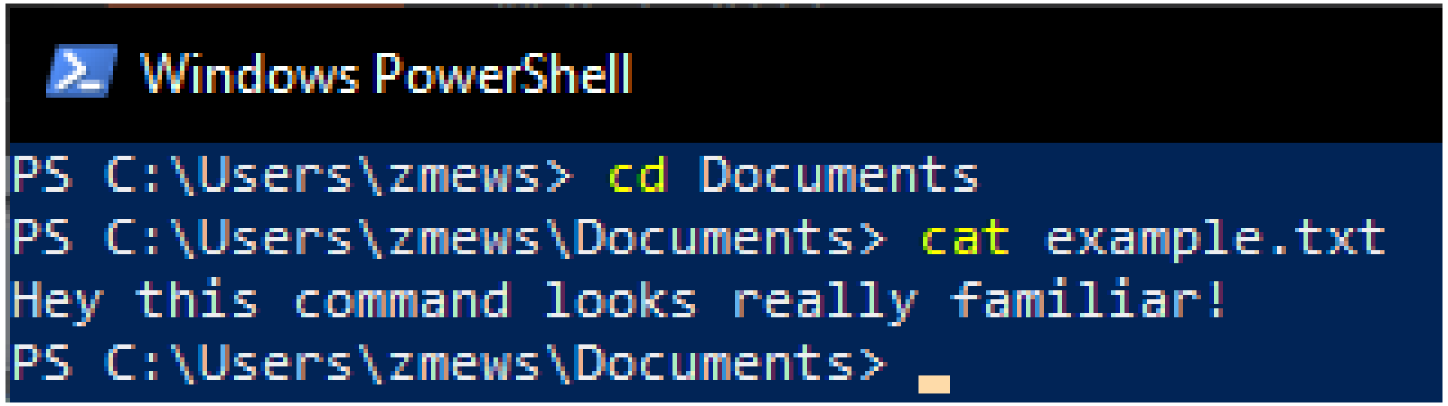


```
Microsoft-User Experience Virtualization-Agent Driver%4Operational.evtx
Microsoft-User Experience Virtualization-App Agent%4Operational.evtx
Microsoft-User Experience Virtualization-IPC%4Operational.evtx
Microsoft-User Experience Virtualization-SQM Uploader%4Operational.evtx
Microsoft-Windows-AAD%4Operational.evtx
Microsoft-Windows-AllJoyn%4Operational.evtx
Microsoft-Windows-All-User-Install-Agent%4Admin.evtx
Microsoft-Windows-AppHost%4Admin.evtx
Microsoft-Windows-AppID%4Operational.evtx
Microsoft-Windows-ApplicabilityEngine%4Operational.evtx
Microsoft-Windows-Application Server-Applications%4Admin.evtx
Microsoft-Windows-Application Server-Applications%4Operational.evtx
```

PowerShell

- PowerShell is object oriented.

- Many Linux commands work by default!
  - **cat**
  - **cd**
  - **ls**
  - **man**
  - **ping**
  - **ssh**
  - **And more!**

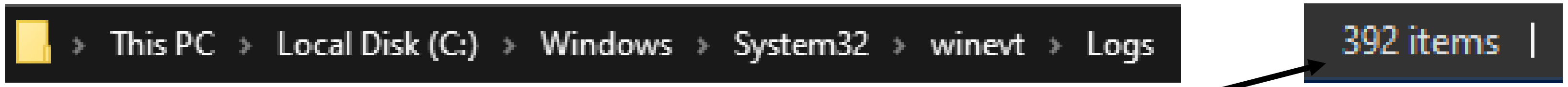


```
PS C:\Users\zmews> cd Documents
PS C:\Users\zmews\Documents> cat example.txt
Hey this command looks really familiar!
PS C:\Users\zmews\Documents> _
```

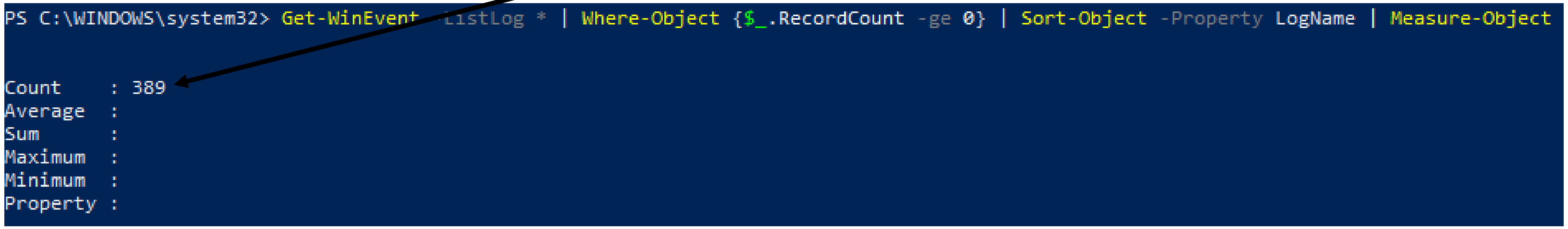
PowerShell

- Windows logs can be queried in PowerShell using the command:
  - **Get-WinEvent**
- Some tags we can combine with Get-WinEvent:
  - **-ListLog**
  - **-LogName**
  - **-ListProvider**
  - **-Path**
- Additional PowerShell functions and tags we can use to manipulate data:
  - **Sort-Object**
  - **Measure-Object**
  - **Where-Object**
  - **-Property**

PowerShell



This PC › Local Disk (C:) › Windows › System32 › winevt › Logs
392 items



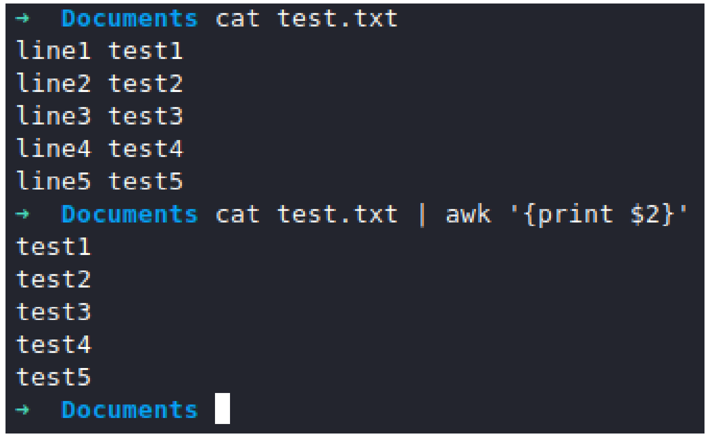
```
PS C:\WINDOWS\system32> Get-WinEvent -ListLog * | Where-Object {$_.RecordCount -ge 0} | Sort-Object -Property LogName | Measure-Object


Count    : 389
Average  :
Sum      :
Maximum  :
Minimum  :
Property :
```

Bash Scripting

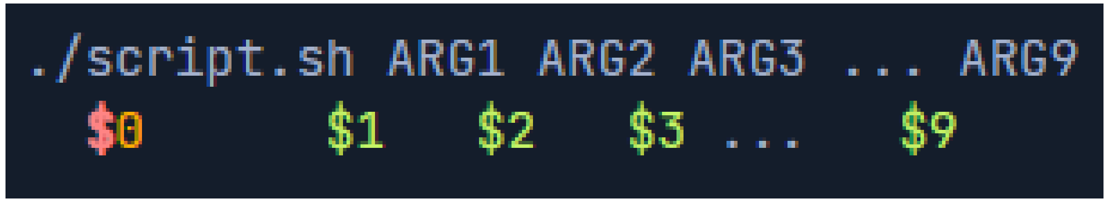
- A must learn for anyone looking to pursue cybersecurity professionally
- A command interpreter AND a programming language.

Bash Scripting

- Commands can be invoked just as if it were the command line.
- All files start with a “Shebang”: **#!/bin/bash**.
- Used to specify the interpreter path (**/bin/bash**).
- **$0-$9** are reserved for arguments. Remember awk?



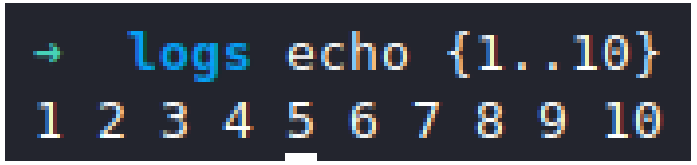
```
→  Documents cat test.txt
line1 test1
line2 test2
line3 test3
line4 test4
line5 test5
→  Documents cat test.txt | awk '{print $2}'
test1
test2
test3
test4
test5
→  Documents
```



```
./script.sh  ARG1  ARG2  ARG3  ...  ARG9
  $0          $1    $2    $3  ...    $9
```

Bash Scripting

- Variables are declared normally but called with **$**
- Note: **$** is not used with making an assignment i.e., **x+=4**
- Sequences are built with curly brackets **{}** separated by **2 periods**.

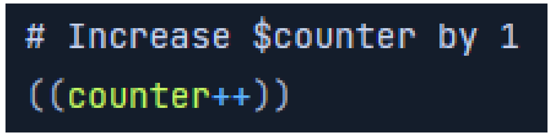


```
→  logs echo {1..10}
1 2 3 4 5 6 7 8 9 10
```

- Strings, integers, files, and Booleans all have different comparison operators i.e., **==**, **-eq**, **-e**, **&&** respectively.

Bash Scripting

- double parentheses is used to specify arithmetic **(())**
- Syntax of loops for loops:



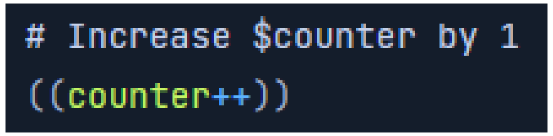
```
# Increase $counter by 1
((counter++))
```

```
for $variable in 1 2 3 4
do
    echo $variable
done
```

```
for $variable in file1 file2 file3
do
        echo $variable
done
```

Bash Scripting

- double parentheses is used to specify arithmetic **(())**
- Syntax of loops for loops:



```
# Increase $counter by 1
((counter++))
```

```
for variable in 1 2 3 4
do
      echo $variable
done
```

```
for variable in file1 file2 file3
do
            echo $variable
done
```

Live Demo

Useful Resources

- Linux Text Processing: https://tldp.org/LDP/abs/html/textproc.html
- Fake Apache Log Generator: https://github.com/kiritbasu/Fake-Apache-Log-Generator
- Microsoft's own documentation for PowerShell is pretty solid.

Proud Sponsors

VOLGENAU SCHOOL
MASON
ENGINEERING

CACI
EVER VIGILANT

WarCollar
industries