Project Title: Windows Log Analysis

Objectives: Analyzing windows event logs to monitor security operations and detect potential threats.

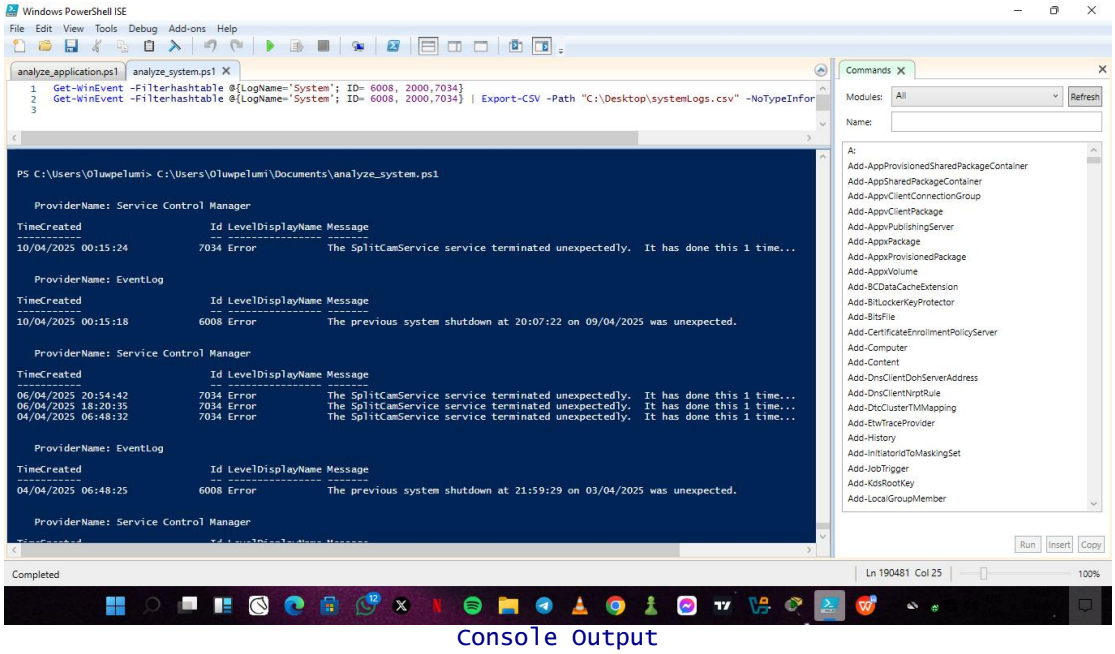
Introduction: Windows Event Logs are activity recorded or changes made to the computer, there are five types of event logs which are;
System event log: which contains events logged by windows system components.
Application event log: contains events logged by application or programs.
Setup event log: contains events related to application setup and installations
Forwarded Event Log: contains events forwarded from another computer.

Tools and Setup: Windows PowerShell (to write the commands script) Microsoft Excel or WPS(to view the .xlxs file)

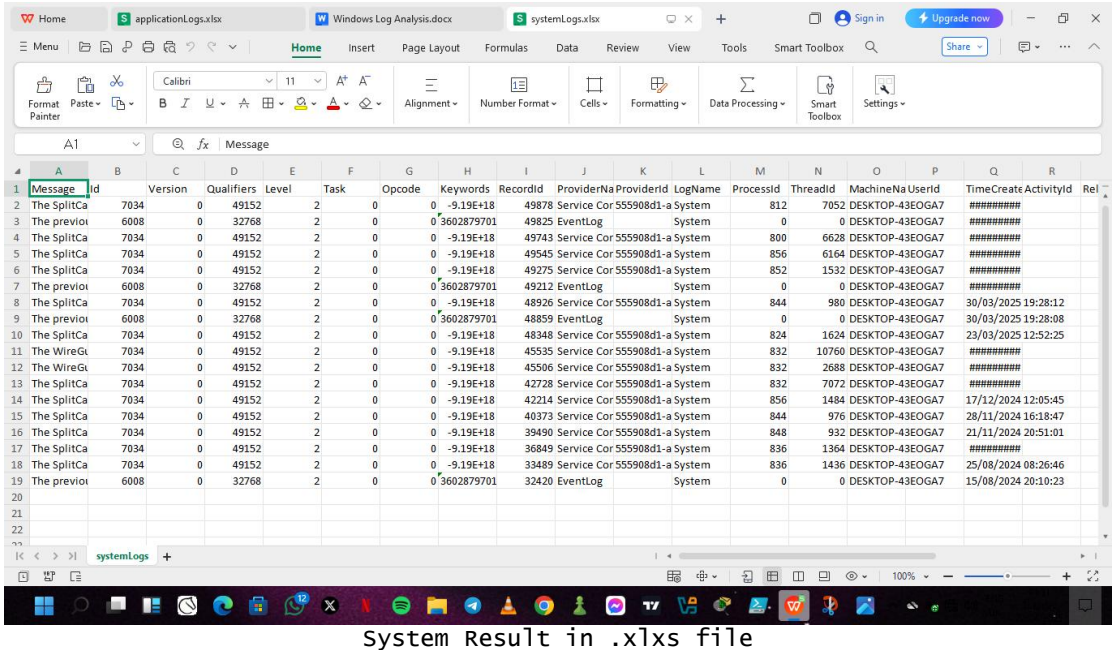
Log Analysis

1. System Event Log: I extracted recent events(e.g service shutdowns, hardware or driver failures) by using event ID which are 7034,6008, 2000 respectively, after formated in a table view then export my results to Systemlogs.xlxs. Scripts;

```
 Get-WinEvent -Filterhashtable @{LogName='System'; ID= 6008, 2000,7034}
Get-WinEvent -Filterhashtable @{LogName='System'; ID= 6008, 2000,7034} |
Export-CSV -Path "C:\Desktop\systemLogs.csv" -NoTypeInformation
```



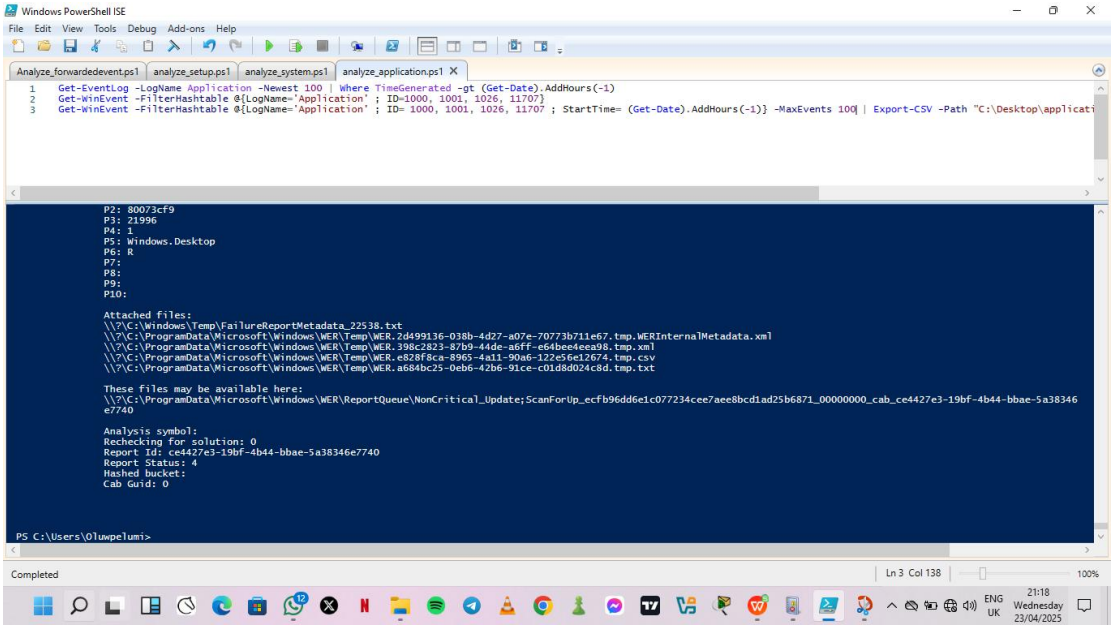
Console Output



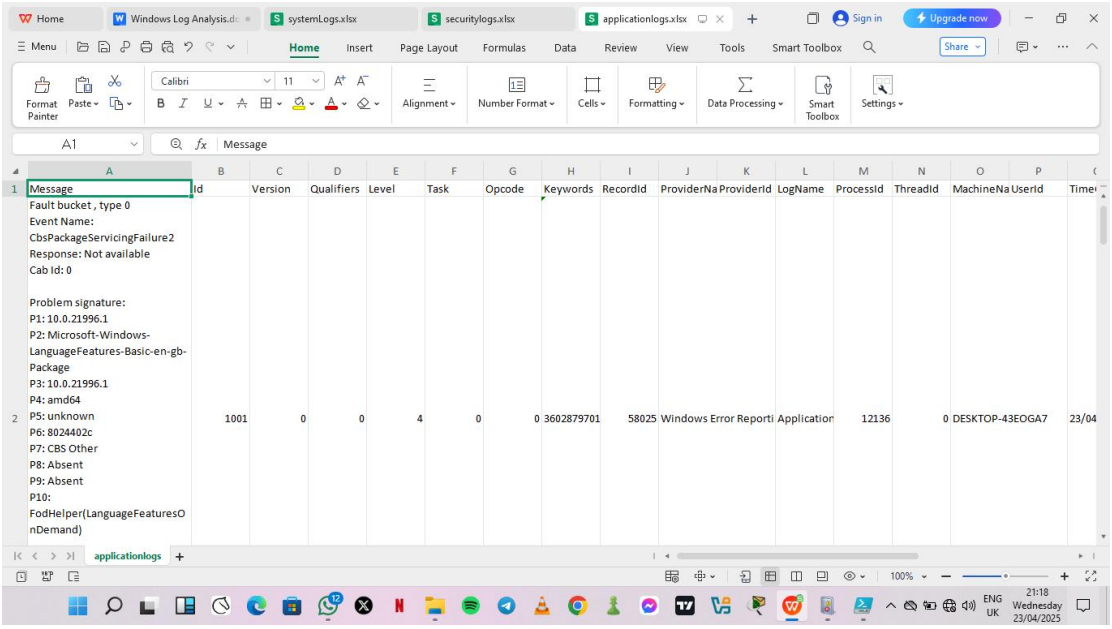
System Result in .xlxs file

2. Application Event: I queried the last 100 events from the past hour and identified application crash event like Event ID 1000, then exported my results to applicationlogs.xlxs. Scripts;

```
 Get-EventLog -LogName Application -Newest 100 | Where TimeGenerated -gt (Get-Date).AddHours(-1)
 Get-WinEvent -FilterHashtable @{LogName='Application' ; ID=1000, 1001, 1026, 11707}
 Get-WinEvent -FilterHashtable @{LogName='Application' ; ID= 1000, 1001, 1026, 11707 ; StartTime= (Get-Date).AddHours(-1)} -MaxEvents 100 | Export-CSV -Path "C:\Desktop\applicationlogs.csv" -NoTypeInformation
```



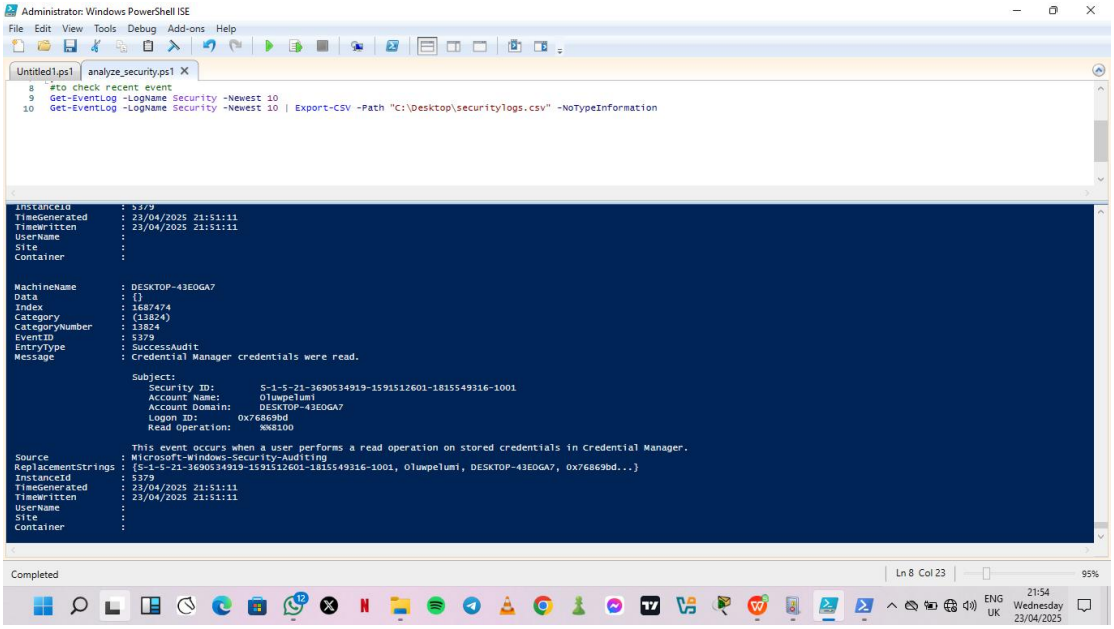
Console Output



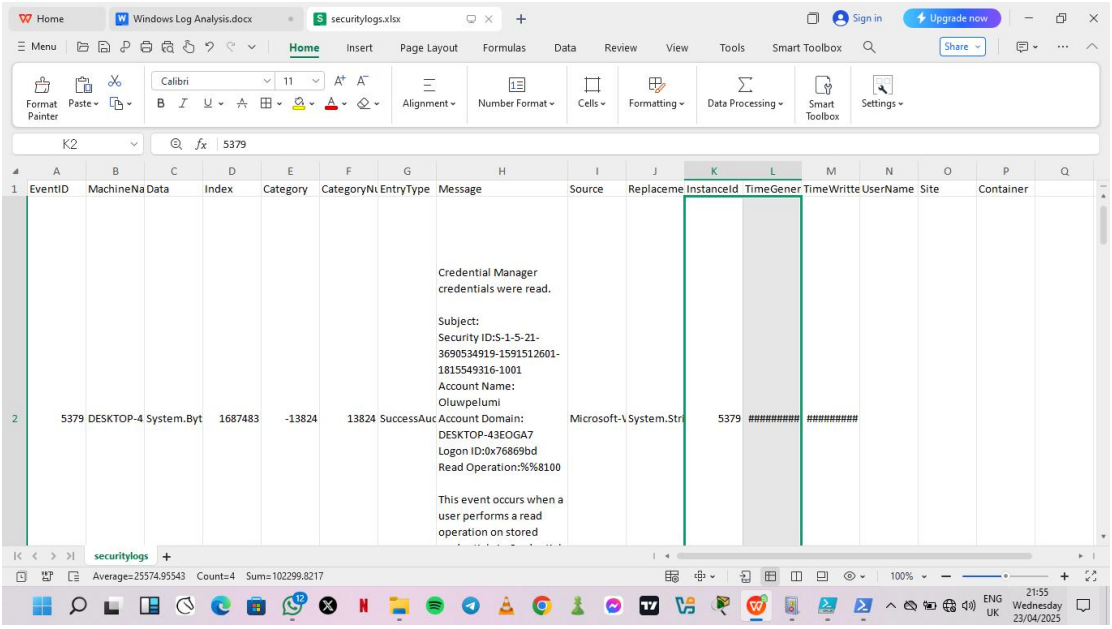
Application Result in .xlxs

3. Security: I detected event IDs for login attempts; 4624(success) & 4625(failure) also handle administrative access within the script, exported result to security.xlxs then simulated a login attempts to test scripts accuracy (used ctrl + L to lock my pc then login again and check for the event log) scripts;

```
Get-WinEvent -FilterHashtable @{LogName='Security' ; ID= 4624 , 4625}
if (-not ([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole(`
    [Security.Principal.WindowsBuiltInRole] "Administrator"))
{
    Write-Warning "This script needs to be run as Administrator!"
    exit
}
Get-EventLog -LogName Security -Newest 10
Get-EventLog -LogName Security -Newest 10 | Export-CSV -Path
"C:\Desktop\securitylogs.csv" -NoTypeInformation
```



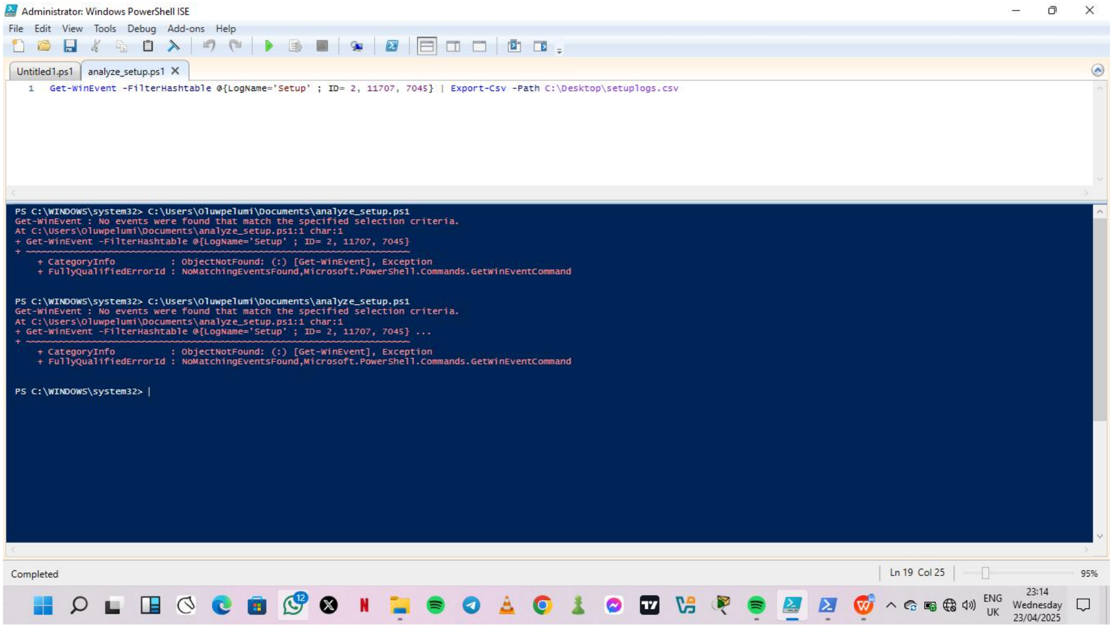
Console Output



Security Results in .xlxs

4. Setup Event: Detect system updates, installations, and configuration events, Event iDs are 2, 11707, 7045 respectively and export results to setuplogs.xlxs. Scripts;

```
Get-WinEvent -FilterHashtable @{LogName='Setup' ; ID= 2, 11707, 7045}
```



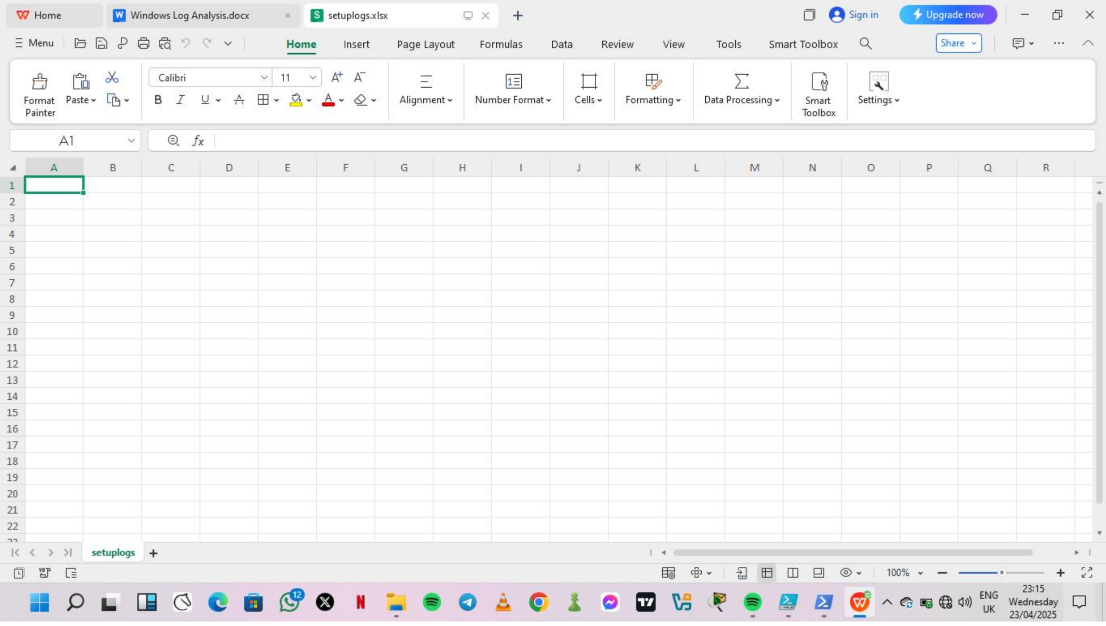
```
Get-WinEvent -FilterHashtable @{LogName='Setup' ; ID= 2, 11707, 7045} | Export-Csv -Path C:\Desktop\setuplogs.csv
```

```
PS C:\WINDOWS\system32> C:\Users\Oluwpelumi\Documents\analyze_setup.ps1
Get-WinEvent : No events were found that match the specified selection criteria.
At C:\Users\Oluwpelumi\Documents\analyze_setup.ps1:1 char:1
+ Get-WinEvent -FilterHashtable @{LogName='Setup' ; ID= 2, 11707, 7045}
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : NoMatchingEventsFound,Microsoft.PowerShell.Commands.GetWinEventCommand

PS C:\WINDOWS\system32> C:\Users\Oluwpelumi\Documents\analyze_setup.ps1
Get-WinEvent : No events were found that match the specified selection criteria.
At C:\Users\Oluwpelumi\Documents\analyze_setup.ps1:1 char:1
+ Get-WinEvent -FilterHashtable @{LogName='Setup' ; ID= 2, 11707, 7045} ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : NoMatchingEventsFound,Microsoft.PowerShell.Commands.GetWinEventCommand

PS C:\WINDOWS\system32> |
```

Console Output



Setup Result in .xlxs(It is empty because there is no events ID recorded in the Setup log)

5. Forwarded Event: My forwarded event log is disabled, to activate it. I need an extra computer to forward event logs.

Summary of Findings

System Log:

- Detected instances of unexpected shutdowns (Event ID 6008).
- Logged service crashes and restarts (Event ID 7034).
- Some hardware-related warnings and errors detected (e.g., Event ID 2000).

Application Log:

- Successful and failed software installations were observed (Event ID 11707, 11724).
- A few application crashes detected (Event ID 1000, 1001).

Security Log:

- Normal logon activity recorded (Event ID 4624).
- No unusual or excessive failed login attempts (Event ID 4625) found within the checked period.

Setup Log:

- No recent system updates or setup-related events detected.
- Possible that updates are managed externally, or the log has been cleared.

Forwarded Events:

- Log is operational but not receiving live forwarded events due to standalone environment.

Recommendations

- Enable Auditing Policies: Ensure all relevant security auditing (logon, privilege use) is enabled via AuditPol.
- Regular Log Reviews: Implement scheduled log reviews or alerts for critical Event IDs (6008, 7034, 4625).
- Retention Policy: Verify log retention settings to avoid overwriting important events.
- Forwarding Setup: For production use, implement Windows Event Forwarding with multiple systems for centralized monitoring.
- Update Confirmation: Check Windows Update settings or group policy to confirm update logs are being written properly.

Conclusion

The event log analysis confirms that the system is generally stable with normal activity in logs. Some critical and warning events were identified, but no evidence of system compromise or misconfiguration was found. Future enhancements such as centralized log monitoring, alerting, and full audit policy coverage are recommended for proactive system health and security tracking.