Useful Windows Event Logs ID's

- **Windows System Logs**
  - Event ID 1074 `(System Shutdown/Restart)` : This event log indicates when and why the system was shut down or restarted. By monitoring these events, you can determine if there are unexpected shutdowns or restarts, potentially revealing malicious activity such as malware infection or unauthorized user access.
  - Event ID 6005 `(The Event log service was started)` : This event log marks the time when the Event Log Service was started. This is an important record, as it can signify a system boot-up, providing a starting point for investigating system performance or potential security incidents around that period. It can also be used to detect unauthorized system reboots.
  - Event ID 6006 `(The Event log service was stopped)` : This event log signifies the moment when the Event Log Service was stopped. It is typically seen when the system is shutting down. Abnormal or unexpected occurrences of this event could point to intentional service disruption for covering illicit activities.
  - Event ID 6013 `(Windows uptime)` : This event occurs once a day and shows the uptime of the system in seconds. A shorter than expected uptime could mean the system has been rebooted, which could signify a potential intrusion or unauthorized activities on the system.
  - Event ID 7040 `(Service status change)` : This event indicates a change in service startup type, which could be from manual to automatic or vice versa. If a crucial service's startup type is changed, it could be a sign of system tampering.
- **Windows Security Logs**
  - Event ID 1102 `(The audit log was cleared)` : Clearing the audit log is often a sign of an attempt to remove evidence of an intrusion or malicious activity.
  - Event ID 1116 `(Antivirus malware detection)` : This event is particularly important because it logs when Defender detects a malware. A surge in these events could indicate a targeted attack or widespread malware infection.
  - Event ID 1118 `(Antivirus remediation activity has started)` : This event signifies that Defender has begun the process of removing or quarantining detected malware. It's important to monitor these events to ensure that remediation activities are successful.
  - Event ID 1119 `(Antivirus remediation activity has succeeded)` : This event signifies that the remediation process for detected malware has been successful. Regular monitoring of these events will help ensure that identified threats are effectively neutralized.

- Event ID 1120 `(Antivirus remediation activity has failed)`: This event is the counterpart to 1119 and indicates that the remediation process has failed. These events should be closely monitored and addressed immediately to ensure threats are effectively neutralized.
- Event ID 4624 `(Successful Logon)`: This event records successful logon events. This information is vital for establishing normal user behavior. Abnormal behavior, such as logon attempts at odd hours or from different locations, could signify a potential security threat.
- Event ID 4625 `(Failed Logon)`: This event logs failed logon attempts. Multiple failed logon attempts could signify a brute-force attack in progress.
- Event ID 4648 `(A logon was attempted using explicit credentials)`: This event is triggered when a user logs on with explicit credentials to run a program. Anomalies in these logon events could indicate lateral movement within a network, which is a common technique used by attackers.
- Event ID 4656 `(A handle to an object was requested)`: This event is triggered when a handle to an object (like a file, registry key, or process) is requested. This can be a useful event for detecting attempts to access sensitive resources.
- Event ID 4672 `(Special Privileges Assigned to a New Logon)`: This event is logged whenever an account logs on with super user privileges. Tracking these events helps to ensure that super user privileges are not being abused or used maliciously.
- Event ID 4698 `(A scheduled task was created)`: This event is triggered when a scheduled task is created. Monitoring this event can help you detect persistence mechanisms, as attackers often use scheduled tasks to maintain access and run malicious code.
- Event ID 4700 & Event ID 4701 `(A scheduled task was enabled/disabled)`: This records the enabling or disabling of a scheduled task. Scheduled tasks are often manipulated by attackers for persistence or to run malicious code, thus these logs can provide valuable insight into suspicious activities.
- Event ID 4702 `(A scheduled task was updated)`: Similar to 4698, this event is triggered when a scheduled task is updated. Monitoring these updates can help detect changes that may signify malicious intent.
- Event ID 4719 `(System audit policy was changed)`: This event records changes to the audit policy on a computer. It could be a sign that someone is trying to cover their tracks by turning off auditing or changing what events get audited.
- Event ID 4738 `(A user account was changed)`: This event records any changes made to user accounts, including changes to privileges, group memberships, and account settings. Unexpected account changes can be a sign of account takeover or insider threats.

- Event ID 4771 `(Kerberos pre-authentication failed)`: This event is similar to 4625 (failed logon) but specifically for Kerberos authentication. An unusual amount of these logs could indicate an attacker attempting to brute force your Kerberos service.
- Event ID 4776 `(The domain controller attempted to validate the credentials for an account)`: This event helps track both successful and failed attempts at credential validation by the domain controller. Multiple failures could suggest a brute-force attack.
- Event ID 5001 `(Antivirus real-time protection configuration has changed)`: This event indicates that the real-time protection settings of Defender have been modified. Unauthorized changes could indicate an attempt to disable or undermine the functionality of Defender.
- Event ID 5140 `(A network share object was accessed)`: This event is logged whenever a network share is accessed. This can be critical in identifying unauthorized access to network shares.
- Event ID 5142 `(A network share object was added)`: This event signifies the creation of a new network share. Unauthorized network shares could be used to exfiltrate data or spread malware across a network.
- Event ID 5145 `(A network share object was checked to see whether client can be granted desired access)`: This event indicates that someone attempted to access a network share. Frequent checks of this sort might indicate a user or a malware trying to map out the network shares for future exploits.
- Event ID 5157 `(The Windows Filtering Platform has blocked a connection)`: This is logged when the Windows Filtering Platform blocks a connection attempt. This can be helpful for identifying malicious traffic on your network.
- Event ID 7045 `(A service was installed in the system)`: A sudden appearance of unknown services might suggest malware installation, as many types of malware install themselves as services.

Sysmon

- Download Sysmon ==> https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
- then start it ->

```
sysmon.exe -i -accepteula -h md5,sha256,imphash -l -n
```

Event id's sysmon

- Sysmon Event ID 1 - Process Creation: Useful for hunts targeting abnormal parent-child process hierarchies, as illustrated in the first lesson with Process Hacker. It's an event we can use later.
- Sysmon Event ID 2 - A process changed a file creation time: Helpful in spotting "time stomp" attacks, where attackers alter file creation times. Bear in mind, not all such actions signal malicious intent.
- Sysmon Event ID 3 - Network connection: A source of abundant noise since machines are perpetually establishing network connections. We may uncover anomalies, but let's consider other quieter areas first.
- Sysmon Event ID 4 - Sysmon service state changed: Could be a useful hunt if attackers attempt to stop Sysmon, though the majority of these events are likely benign and informational, considering Sysmon's frequent legitimate starts and stops.
- Sysmon Event ID 5 - Process terminated: This might aid us in detecting when attackers kill key processes or use sacrificial ones. For instance, Cobalt Strike often spawns temporary processes like werfault, the termination of which would be logged here, as well as the creation in ID 1.
- Sysmon Event ID 6 - Driver loaded: A potential flag for BYOD (bring your own driver) attacks, though this is less common. Before diving deep into this, let's weed out more conspicuous threats first.
- Sysmon Event ID 7 - Image loaded: Allows us to track dll loads, which is handy in detecting DLL hijacks.
- Sysmon Event ID 8 - CreateRemoteThread: Potentially aids in identifying injected threads. While remote threads can be created legitimately, if an attacker misuses this API, we can potentially trace their rogue process and what they injected into.
- Sysmon Event ID 10 - ProcessAccess: Useful for spotting remote code injection and memory dumping, as it records when handles on processes are made.
- Sysmon Event ID 11 - FileCreate: With many files being created frequently due to updates, downloads, etc., it might be challenging to aim our hunt directly here. However, these events can be beneficial in correlating or identifying a file's origins later.
- Sysmon Event ID 12 - RegistryEvent (Object create and delete) & Sysmon Event ID 13 - RegistryEvent (Value Set): While numerous events take place here, many registry events can be malicious, and with a good idea of what to look for, hunting here can be fruitful.
- Sysmon Event ID 15 - FileCreateStreamHash: Relates to file streams and the "Mark of the Web" pertaining to external downloads, but we'll leave this aside for now.
- Sysmon Event ID 16 - Sysmon config state changed: Logs alterations in Sysmon configuration, useful for spotting tampering.
- Sysmon Event ID 17 - Pipe created & Sysmon Event ID 18 - Pipe connected: Record pipe creations and connections. They can help observe malware's interprocess communication attempts, usage of PsExec, and SMB lateral movement.

- Sysmon Event ID 22 - DNSEvent: Tracks DNS queries, which can be beneficial for monitoring beacon resolutions and DNS beacons.
- Sysmon Event ID 23 - FileDelete: Monitors file deletions, which can provide insights into whether a threat actor cleaned up their malware, deleted crucial files, or possibly attempted a ransomware attack.
- Sysmon Event ID 25 - ProcessTampering (Process image change): Alerts on behaviors such as process herpadering, acting as a mini AV alert filter.

ETW

Useful Providers

- `Microsoft-Windows-Kernel-Process`: This ETW provider is instrumental in monitoring process-related activity within the Windows kernel. It can aid in detecting unusual process behaviors such as process injection, process hollowing, and other tactics commonly used by malware and advanced persistent threats (APTs).
- `Microsoft-Windows-Kernel-File`: As the name suggests, this provider focuses on file-related operations. It can be employed for detection scenarios involving unauthorized file access, changes to critical system files, or suspicious file operations indicative of exfiltration or ransomware activity.
- `Microsoft-Windows-Kernel-Network`: This ETW provider offers visibility into network-related activity at the kernel level. It's especially useful in detecting network-based attacks such as data exfiltration, unauthorized network connections, and potential signs of command and control (C2) communication.
- `Microsoft-Windows-SMBClient/SMBServer`: These providers monitor Server Message Block (SMB) client and server activity, providing insights into file sharing and network communication. They can be used to detect unusual SMB traffic patterns, potentially indicating lateral movement or data exfiltration.
- `Microsoft-Windows-DotNETRuntime`: This provider focuses on .NET runtime events, making it ideal for identifying anomalies in .NET application execution, potential exploitation of .NET vulnerabilities, or malicious .NET assembly loading.
- `OpenSSH`: Monitoring the OpenSSH ETW provider can provide important insights into Secure Shell (SSH) connection attempts, successful and failed authentications, and potential brute force attacks.
- `Microsoft-Windows-VPN-Client`: This provider enables tracking of Virtual Private Network (VPN) client events. It can be useful for identifying unauthorized or suspicious VPN connections.

- `Microsoft-Windows-PowerShell`: This ETW provider tracks PowerShell execution and command activity, making it invaluable for detecting suspicious PowerShell usage, script block logging, and potential misuse or exploitation.
- `Microsoft-Windows-Kernel-Registry`: This provider monitors registry operations, making it useful for detection scenarios related to changes in registry keys, often associated with persistence mechanisms, malware installation, or system configuration changes.
- `Microsoft-Windows-CodeIntegrity`: This provider monitors code and driver integrity checks, which can be key in identifying attempts to load unsigned or malicious drivers or code.
- `Microsoft-Antimalware-Service`: This ETW provider can be employed to detect potential issues with the antimalware service, including disabled services, configuration changes, or potential evasion techniques employed by malware.
- `WinRM`: Monitoring the Windows Remote Management (WinRM) provider can reveal unauthorized or suspicious remote management activity, often indicative of lateral movement or remote command execution.
- `Microsoft-Windows-TerminalServices-LocalSessionManager`: This provider tracks local Terminal Services sessions, making it useful for detecting unauthorized or suspicious remote desktop activity.
- `Microsoft-Windows-Security-Mitigations`: This provider keeps tabs on the effectiveness and operations of security mitigations in place. It's essential for identifying potential bypass attempts of these security controls.
- `Microsoft-Windows-DNS-Client`: This ETW provider gives visibility into DNS client activity, which is crucial for detecting DNS-based attacks, including DNS tunneling or unusual DNS requests that may indicate C2 communication.
- `Microsoft-Antimalware-Protection`: This provider monitors the operations of antimalware protection mechanisms. It can be used to detect any issues with these mechanisms, such as disabled protection features, configuration changes, or signs of evasion techniques employed by malicious actors.

Get providers :

```
logman.exe query providers | findstr "Process"
```

ETW with the provider logging

```
SilkETW.exe -t user -pn Microsoft-Windows-Kernel-Process -ot file -p
C:\windows\temp\etw.json
```

Get-WinEvent

List Logs

```
Get-WinEvent -ListLog * | Select-Object LogName, RecordCount, IsClassicLog,
IsEnabled, LogMode, LogType | Format-Table -AutoSize
```

List Providers

```
Get-WinEvent -ListProvider * | Format-Table -AutoSize
```

interact with Events

```
Get-WinEvent -LogName 'System' -MaxEvents 50 | Select-Object TimeCreated,
ID, ProviderName, LevelDisplayName, Message | Format-Table -AutoSize
```

Search in Event

```
Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-
Sysmon/Operational'; ID=1,3} | Select-Object TimeCreated, ID, ProviderName,
LevelDisplayName, Message | Format-Table -AutoSize
```

search with date

```
PS C:\Users\Administrator> $startDate = (Get-Date -Year 2023 -Month 5 -Day
28).Date
 PS C:\Users\Administrator> $endDate   = (Get-Date -Year 2023 -Month 6 -Day
3).Date
 PS C:\Users\Administrator> Get-WinEvent -FilterHashtable
@{LogName='Microsoft-Windows-Sysmon/Operational'; ID=1,3;
StartTime=$startDate; EndTime=$endDate} | Select-Object TimeCreated, ID,
ProviderName, LevelDisplayName, Message | Format-Table -AutoSize
```