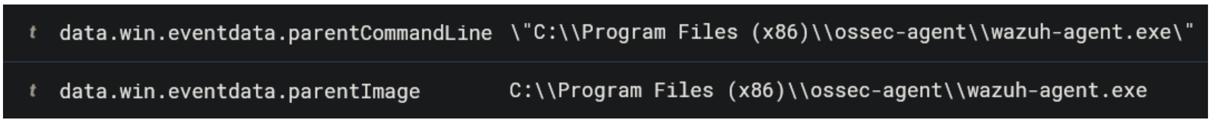
Event Analysis: PowerShell Usage

These analysed events occurred when PowerShell was invoked to create a new process. The most important fields were ParentImage (the executable that invoked PowerShell) and parentCommandLine (the command that PowerShell was used to create a process).

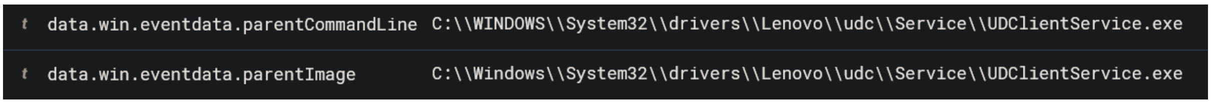
- **Wazuh Agent (50):** ParentImage was the Wazuh agent executable with the CommandLine being the file path to the executable. This likely shows the Wazuh agent started monitoring



```
data.win.eventdata.parentCommandLine  \"C:\\Program Files (x86)\\ossec-agent\\wazuh-agent.exe\"
data.win.eventdata.parentImage        C:\\Program Files (x86)\\ossec-agent\\wazuh-agent.exe
```

Figure 1: Example parentCommandLine/parentImage Fields for a Wazuh Agent Event

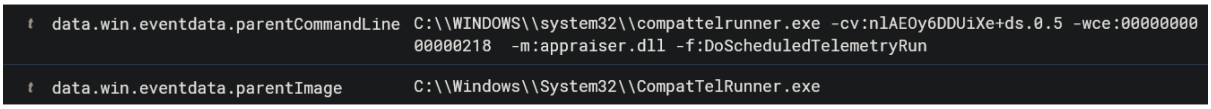
- **Lenovo Driver Service (UDClientService.exe) (59):** ParentImage was nested in the file path \drivers\Lenovo and the CommandLine was the path to UDClientService.exe. This likely enables Lenovo drivers to function properly



```
data.win.eventdata.parentCommandLine  C:\\WINDOWS\\System32\\drivers\\Lenovo\\udc\\Service\\UDClientService.exe
data.win.eventdata.parentImage        C:\\Windows\\System32\\drivers\\Lenovo\\udc\\Service\\UDClientService.exe
```

Figure 2: Example parentCommandLine/parentImage Fields for a Lenovo Driver Service Event

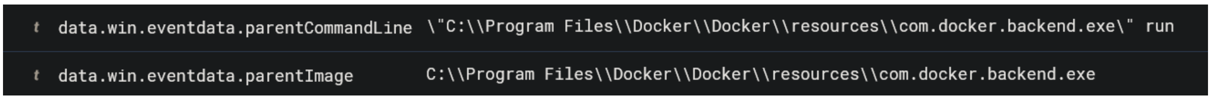
- **OS Processes (79):** ParentImage file path shows the executable is from the System32 folder and the commands cover a wide range. These are likely privileged OS processes, e.g. one example is a scheduled Telemetry run



```
data.win.eventdata.parentCommandLine  C:\\WINDOWS\\system32\\compattelrunner.exe -cv:nlAEOy6DDUiXe+ds.0.5 -wce:00000000
                                      00000218  -m:appraiser.dll -f:DoScheduledTelemetryRun
data.win.eventdata.parentImage        C:\\Windows\\System32\\CompatTelRunner.exe
```

Figure 3: Example parentCommandLine/parentImage Fields for an OS Process Event

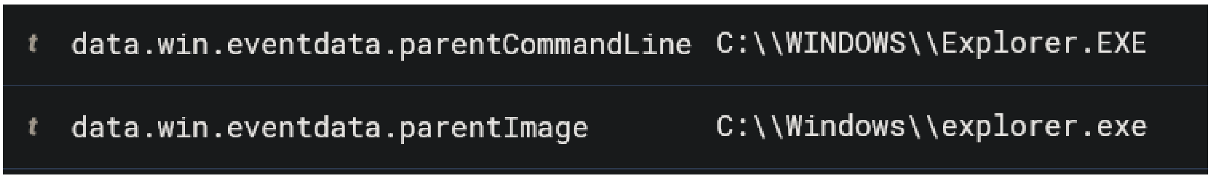
- **Docker (16):** ParentImage is docker backend, CommandLine is always the file path followed by run. This likely used for Docker container initialisation.



```
data.win.eventdata.parentCommandLine  \"C:\\Program Files\\Docker\\Docker\\resources\\com.docker.backend.exe\" run
data.win.eventdata.parentImage        C:\\Program Files\\Docker\\Docker\\resources\\com.docker.backend.exe
```

Figure 4: Example parentCommandLine/parentImage Fields for a Docker Event

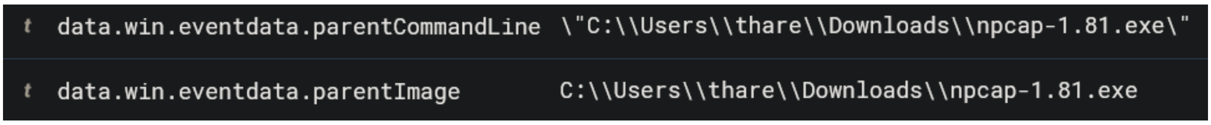
- **Explorer (8):** ParentImage is Explorer.exe and so is the CommandLine, likely for files that require admin permissions to access via a UAC prompt



```
t  data.win.eventdata.parentCommandLine  C:\\WINDOWS\\Explorer.EXE
t  data.win.eventdata.parentImage        C:\\Windows\\explorer.exe
```

Figure 5: Example parentCommandLine/parentImage Fields for a Docker Event

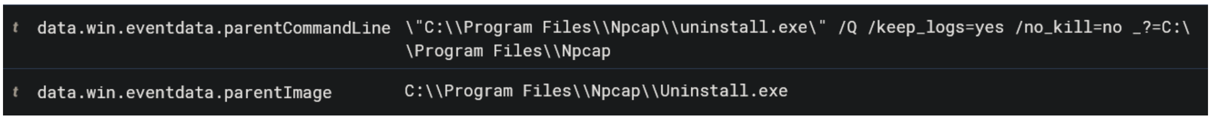
- **Ncap (6):** ParentImage is Ncap and CommandLine is the Ncap.exe file path. This is likely from packet capture while using Suricata



```
t  data.win.eventdata.parentCommandLine  \"C:\\Users\\thare\\Downloads\\npcap-1.81.exe\"
t  data.win.eventdata.parentImage        C:\\Users\\thare\\Downloads\\npcap-1.81.exe
```

Figure 6: Example parentCommandLine/parentImage Fields for a Ncap Event

- **Ncap Uninstall (1):** ParentImage is uninstall.exe within the Ncap folder. This uses a PowerShell command for the uninstall process



```
t  data.win.eventdata.parentCommandLine  \"C:\\Program Files\\Npcap\\uninstall.exe\" /Q /keep_logs=yes /no_kill=no _?=C:\
                                         \Program Files\\Npcap
t  data.win.eventdata.parentImage        C:\\Program Files\\Npcap\\Uninstall.exe
```

Figure 7: Example parentCommandLine/parentImage Fields for a Ncap uninstall Event

Most processes tend to use PowerShell to run themselves, usually via running a .exe through its file path in these examples. In the context of personal SIEM application, understanding legitimate PowerShell usage helps establish baseline normal activity, which allows for recognition of suspicious activity or unauthorised behaviour.