**How are Windows Services Leveraged by Hackers/ Malicious Actors**

* Breached Windows Services allow for a malicious actor to gain and maintain persistence in the target system.
  + Bad Actors can install new services or modify existing ones to execute at startup to maintain persistence on the system. This can be done by modifying the *sc.exe* or other system utilities, directly changing a windows registry, or direct connection with the Windows API.
  + They may also use the services in order to install and execute malicious drivers. A driver file may be pushed into the user’s system and then controlled via Native API functions, creating/modifying registry service values, or command line utilities such as *PnPUtil.exe*.
    - The above may be leveraged as Rootkit drivers to hide themselves from being caught in the target system.
    - When creating and using said malicious services, they will use Masquerading of the task name in order to hide the task or service. The service can also be hidden from view using the technique of hiding artifacts(can set service perms via Security Descriptor Definition Language SDDL), this will hide the windows service from standard command views like *Get-Service, sc query, services.exe*
  + Using the *sc* command malicious actors can also try and avoid detection by disabling

**What are Windows Services, How do they function? How are they triggered? How is Info Stored?**

* Windows Services start at boot time and primarily focus on performing tasks in that background for system function.
* Windows Services configuration info (file path to services executable, recovery programs and/or commands) is stored within the *Windows Registry*
* Windows Services are created under the Admin privilege layer, but when executed are under the SYSTEM privilege (Could allow for privilege escalation for malicious actors)
* “The ***HKLM\SYSTEM\CurrentControlSet\Services*** registry tree stores information about each service on the system. Each driver has a key of the form ***HKLM\SYSTEM\CurrentControlSet\Services\***DriverName.”

**Sysinternals, and which ones can help me?**

* Procmon – A real time monitoring tool of all windows processes and key creations, modification, closings. Has plenty of filtering tools in order to narrow down search criteria.
* AutoRuns – Shows what programs and services are configured to start up automatically on boot time. It also shows a full list for registry and file locations where apps may configure auto-start settings.
  + “This is *the* most comprehensive auto-start monitoring tool that not only knows of the programs and drivers that are rigged to auto-run on the system via the “Startup” folder and Registry keys (such as Run, RunOnce, and others), but *also*reports Explorer shell extensions, toolbars, browser helper objects, “winlogon” notifications, auto-start services, and *a lot*more.”
* ListDLLs – Lists all the currently loaded DLL’s on the system and their information. It also shows their location path, version numbers and digital signatures. Can scan for unsigned DLL’s
* PendMoves – Lists all file rename and delete commands that are set to execute at the next boot cycle.
  + Could maybe be used to detect DLL’s or driver files that a malicious actor is trying to delete next boot.
* ProcExp – “A process “handle” is an integer value that helps Windows identify a process. This tool shows information regarding processes having any handles, or DLLs, or files, or Registry keys, or other objects either opened or loaded, and more.”
* PsExec – Executes processes on remote systems
  + “This is a lightweight remote administration tool that allows to execute processes on remote systems *without* the need of any additional client installations on the remote system (such as Telnet or Remote Desktop). It also supports full interactivity for console-based applications as it effectively redirects inputs and outputs to and from these console-based applications.”
  + This tool is also used for executing commands—on a remote system—that otherwise are fundamentally not intended for execution against remote systems (such as ipconfig).
  + Can also use PsFile and PsList in order to show files open on remote systems
* SigCheck – “This tools dumps file information of a specified file to show the file version number, digital signature information, certificate chains, and an optional status-check of that file on **[VirusTotal](https://www.virustotal.com/" \t "_blank)** (performs automated file scanning against over 70 anti-malware engines).”
* Sysmon – Monitors and reports essential system activities into the Windows Event Log
  + “This tool is a system service and device driver (hence, kernel-level) that persists across system reboots once installed, and then monitors and logs system activity into the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation times.”

**Windows Event Log / Windows Registry and useful things to watch for**

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

A white text on a white background

Description automatically generated

**Sources:**[**https://attack.mitre.org/techniques/T1543/003/**](https://attack.mitre.org/techniques/T1543/003/)

[**https://live.sysinternals.com/**](https://live.sysinternals.com/) **(this is more for me later)**

[**https://learn.microsoft.com/en-us/windows-hardware/drivers/install/hklm-system-currentcontrolset-services-registry-tree**](https://learn.microsoft.com/en-us/windows-hardware/drivers/install/hklm-system-currentcontrolset-services-registry-tree)

[**https://learn.microsoft.com/en-us/sysinternals/downloads/procmon**](https://learn.microsoft.com/en-us/sysinternals/downloads/procmon)

[**https://learn.microsoft.com/en-us/dotnet/framework/windows-services/introduction-to-windows-service-applications**](https://learn.microsoft.com/en-us/dotnet/framework/windows-services/introduction-to-windows-service-applications)

[**https://thecomputernoob.com/2021/10/06/sysinternals-pstools-bible/**](https://thecomputernoob.com/2021/10/06/sysinternals-pstools-bible/)

[**https://car.mitre.org/analytics/CAR-2021-01-007/**](https://car.mitre.org/analytics/CAR-2021-01-007/)