Create or Modify System Process: Windows Service

Other sub-techniques of Create or Modify System Process (5)

Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions.[1] Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry.

Adversaries may install a new service or modify an existing service to execute at startup in order to persist on a system. Service configurations can be set or modified using system utilities (such as sc.exe), by directly modifying the Registry, or by interacting directly with the Windows API.

Adversaries may also use services to install and execute malicious drivers. For example, after dropping a driver file (ex: `.sys`) to disk, the payload can be loaded and registered via Native API functions such as `CreateServiceW()` (or manually via functions such as `ZwLoadDriver()` and `ZwSetValueKey()`), by creating the required service Registry values (i.e. Modify Registry), or by using command-line utilities such as `PnPUtil.exe`.[2][3][4] Adversaries may leverage these drivers as Rootkits to hide the presence of malicious activity on a system. Adversaries may also load a signed yet vulnerable driver onto a compromised machine (known as "Bring Your Own Vulnerable Driver" (BYOVD)) as part of Exploitation for Privilege Escalation.[5][4]

Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges. Adversaries may also directly start services through Service Execution.

To make detection analysis more challenging, malicious services may also incorporate Masquerade Task or Service (ex: using a service and/or payload name related to a legitimate OS or benign software component). Adversaries may also create 'hidden' services (i.e., Hide Artifacts), for example by using the `sc sdset` command to set service permissions via the Service Descriptor Definition Language (SDDL). This may hide a Windows service from the view of standard service enumeration methods such as `Get-Service`, `sc query`, and `services.exe`.[6][7]

ID: T1543.003

Sub-technique of: T1543

Tactics: Persistence, Privilege Escalation

Platforms: Windows

Effective Permissions: Administrator, SYSTEM

Contributors: Akshat Pradhan, Qualys; Matthew Demaske, Adaptforward; Mayuresh Dani, Qualys; Pedro Harrison; Wietze Beukema, @wietze; Wirapong Petshagun

Version: 1.5

Created: 17 January 2020

Last Modified: 11 April 2024

Version Permalink

Procedure Examples

ID	Name	Description
C0025	2016 Ukraine Electric Power Attack	During the 2016 Ukraine Electric Power Attack, Sandworm Team used an arbitrary system service to load at system boot for persistence for Industroyer. They also replaced the ImagePath registry value of a Windows service with a new backdoor binary. [8]
G1030	Agrius	Agrius has deployed IPsec Helper malware post-exploitation and registered it as a service for persistence.[9]
S0504	Anchor	Anchor can establish persistence by creating a service.[10]
S0584	AppleJeus	AppleJeus can install itself as a service.[11]
G0073	APT19	An APT19 Port 22 malware variant registers itself as a service.[12]
G0022	APT3	APT3 has a tool that creates a new service for persistence.[13]
G0050	APT32	APT32 modified Windows Services to ensure PowerShell scripts were loaded on the system. APT32 also creates a Windows service to establish persistence.[14][15][16]
G0082	APT38	APT38 has installed a new Windows service to establish persistence.[17]
G0096	APT41	APT41 modified legitimate Windows services to install malware backdoors.[18][19] APT41 created the StorSyncSvc service to provide persistence for Cobalt Strike.[20]
C0040	APT41 DUST	APT41 DUST used Windows Services with names such as `Windows Defend` for persistence of DUSTPAN.[21]
G0143	Aquatic Panda	Aquatic Panda created new Windows services for persistence that masqueraded as legitimate Windows services via name change.[22]
S0438	Attor	Attor's dispatcher can establish persistence by registering a new service.[23]
S0347	AuditCred	AuditCred is installed as a new service on the system.[24]
S0239	Bankshot	Bankshot can terminate a specific process by its process id.[25][26]
S0127	BBSRAT	BBSRAT can modify service configurations.[27]
S0268	Bisonal	Bisonal has been modified to be used as a Windows service.[28]
S0570	BitPaymer	BitPaymer has attempted to install itself as a service to maintain persistence.[29]
S1070	Black Basta	Black Basta can create a new service to establish persistence.[30][31]
S0089	BlackEnergy	One variant of BlackEnergy creates a new service using either a hard-coded or randomly generated name.[32]
G0108	Blue Mockingbird	Blue Mockingbird has made their XMRIG payloads persistent as a Windows Service.[33]
S0204	Briba	Briba installs a service pointing to a malicious DLL dropped to disk.[34]
G0008	Carbanak	Carbanak malware installs itself as a service to provide persistence and SYSTEM privileges.[35]
S0335	Carbon	Carbon establishes persistence by creating a service and naming it based off the operating system version running on the current machine.[36]
S0261	Catchamas	Catchamas adds a new service named NetAdapter to establish persistence.[37]

ID	Name	Description
G1021	Cinnamon Tempest	Cinnamon Tempest has created system services to establish persistence for deployed tooling.[38]
S0660	Clambling	Clambling can register itself as a system service to gain persistence.[39]
G0080	Cobalt Group	Cobalt Group has created new services to establish persistence.[40]
S0154	Cobalt Strike	Cobalt Strike can install a new service.[41]
S0608	Conficker	Conficker copies itself into the `%systemroot%\system32` directory and registers as a service.[42]
S0050	CosmicDuke	CosmicDuke uses Windows services typically named "javamtsup" for persistence.[43]
S0046	CozyCar	One persistence mechanism used by CozyCar is to register itself as a Windows service.[44]
S0625	Cuba	Cuba can modify services by using the `OpenService` and `ChangeServiceConfig` functions.[45]
G0105	DarkVishnya	DarkVishnya created new services for shellcode loaders distribution.[46]
S1033	DCSrv	DCSrv has created new services for persistence by modifying the Registry.[47]
S0567	Dtrack	Dtrack can add a service called WBService to establish persistence.[48]
S0038	Duqu	Duqu creates a new service that loads a malicious driver when the system starts. When Duqu is active, the operating system believes that the driver is legitimate, as it has been signed with a valid private key.[49]
S1158	DUSTPAN	DUSTPAN can persist as a Windows Service in operations.[21]
S0024	Dyre	Dyre registers itself as a service by adding several Registry keys.[50]
G1006	Earth Lusca	Earth Lusca created a service using the command `sc create "SysUpdate" binpath= "cmd /c start "[file path]""&&sc config "SysUpdate" start= auto&&netstart SysUpdate` for persistence.[51]
S0081	Elise	Elise configures itself as a service.[52]
S0082	Emissary	Emissary is capable of configuring itself as a service.[53]
S0367	Emotet	Emotet has been observed creating new services to maintain persistence.[54][55][56]
S0363	Empire	Empire can utilize built-in modules to modify service binaries and restore them to their original state.[57]
S0343	Exaramel for Windows	The Exaramel for Windows dropper creates and starts a Windows service named wsmprovav with the description "Windows Check AV."[58]
S0181	FALLCHILL	FALLCHILL has been installed as a Windows service.[11]
G0046	FIN7	FIN7 created new Windows services and added them to the startup directories for persistence.[59]
S0182	FinFisher	FinFisher creates a new Windows service with the malicious executable for persistence.[60][61]
S1044	FunnyDream	FunnyDream has established persistence by running `sc.exe` and by setting the `WSearch` service to run automatically.[62]
S0666	Gelsemium	Gelsemium can drop itself in `C:\Windows\System32\spool\prtprocs\x64\winprint.dll` as an alternative Print Processor to be loaded automatically when the spoolsv Windows service starts.[63]
S0032	gh0st RAT	gh0st RAT can create a new service to establish persistence.[64][65]

ID	Name	Description
S0493	GoldenSpy	GoldenSpy has established persistence by running in the background as an autostart service.[66]
S0342	GreyEnergy	GreyEnergy chooses a service, drops a DLL file, and writes it to that serviceDLL Registry key.[67]
S0071	hcdLoader	hcdLoader installs itself as a service for persistence.[68][69]
S0697	HermeticWiper	HermeticWiper can load drivers by creating a new service using the `CreateServiceW` API.[3]
S0203	Hydraq	Hydraq creates new services to establish persistence.[70][71][72]
S0604	Industroyer	Industroyer can use an arbitrary system service to load at system boot for persistence and replaces the ImagePath registry value of a Windows service with a new backdoor binary.[8]
S0259	InnaputRAT	Some InnaputRAT variants create a new Windows service to establish persistence.[73]
S0260	InvisiMole	InvisiMole can register a Windows service named CsPower as part of its execution chain, and a Windows service named clr_optimization_v2.0.51527_X86 to achieve persistence.[5]
S0044	JHUHUGIT	JHUHUGIT has registered itself as a service to establish persistence.[74]
S0265	Kazuar	Kazuar can install itself as a new service.[75]
G0004	Ke3chang	Ke3chang backdoor RoyalDNS established persistence through adding a service called `Nwsapagent`.[76]
S0387	KeyBoy	KeyBoy installs a service pointing to a malicious DLL dropped to disk.[77]
G0094	Kimsuky	Kimsuky has created new services for persistence.[78][79]
S0356	KONNI	KONNI has registered itself as a service using its export function.[80]
S0236	Kwampirs	Kwampirs creates a new service named WmiApSrvEx to establish persistence.[81]
G0032	Lazarus Group	Several Lazarus Group malware families install themselves as new services.[82][83]
S0451	LoudMiner	LoudMiner can automatically launch a Linux virtual machine as a service at startup if the AutoStart option is enabled in the VBoxVmService configuration file.[84]
S0149	MoonWind	MoonWind installs itself as a new service with automatic startup to establish persistence. The service checks every 60 seconds to determine if the malware is running; if not, it will spawn a new instance.[85]
S0205	Naid	Naid creates a new service to establish.[86]
S0630	Nebulae	Nebulae can create a service to establish persistence.[87]
S0210	Nerex	Nerex creates a Registry subkey that registers a new service.[88]
S0118	Nidiran	Nidiran can create a new service named msamger (Microsoft Security Accounts Manager).[89]
S1090	NightClub	NightClub has created a Windows service named `WmdmPmSp` to establish persistence.[90]
S1100	Ninja	Ninja can create the services `httpsvc` and `w3esvc` for persistence .[91]
S0439	Okrum	To establish persistence, Okrum can install itself as a new service named NtmSsvc.[92]
C0012	Operation CuckooBees	During Operation CuckooBees, the threat actors modified the `IKEEXT` and `PrintNotify` Windows services for persistence.[93]

ID	Name	Description
C0006	Operation Honeybee	During Operation Honeybee, threat actors installed DLLs and backdoors as Windows services.[94]
S0664	Pandora	Pandora has the ability to gain system privileges through Windows services.[95]
S1031	PingPull	PingPull has the ability to install itself as a service.[96]
S0501	PipeMon	PipeMon can establish persistence by registering a malicious DLL as an alternative Print Processor which is loaded when the print spooler service starts.[97]
S0013	PlugX	PlugX can be added as a service to establish persistence. PlugX also has a module to change service configurations as well as start, control, and delete services.[98][99][100][101][102]
S0012	PoisonIvy	PoisonIvy creates a Registry subkey that registers a new service. PoisonIvy also creates a Registry entry modifying the Logical Disk Manager service to point to a malicious DLL dropped to disk.[103]
S0194	PowerSploit	PowerSploit contains a collection of Privesc-PowerUp modules that can discover and replace/modify service binaries, paths, and configs.[104][105]
G0056	PROMETHIUM	PROMETHIUM has created new services and modified existing services for persistence.[106]
S0029	PsExec	PsExec can leverage Windows services to escalate privileges from administrator to SYSTEM with the `-s` argument.[107]
S0650	QakBot	QakBot can remotely create a temporary service on a target host.[108]
S0481	Ragnar Locker	Ragnar Locker has used sc.exe to create a new service for the VirtualBox driver.[109]
S0629	RainyDay	RainyDay can use services to establish persistence.[87]
S0169	RawPOS	RawPOS installs itself as a service to maintain persistence.[110][111][112]
S0495	RDAT	RDAT has created a service when it is installed on the victim machine.[113]
S0172	Reaver	Reaver installs itself as a new service.[114]
S0074	Sakula	Some Sakula samples install themselves as services for persistence by calling WinExec with the `net start` argument.[115]
S1099	Samurai	Samurai can create a service at `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost` to trigger execution and maintain persistence.[91]
S0345	Seasalt	Seasalt is capable of installing itself as a service.[116]
S0140	Shamoon	Shamoon creates a new service named "ntssrv" to execute the payload. Newer versions create the "MaintenaceSrv" and "hdv_725x" services.[117][118]
S0444	ShimRat	ShimRat has installed a Windows service to maintain persistence on victim machines.[119]
S0692	SILENTTRINITY	SILENTTRINITY can establish persistence by creating a new service.[120]
S0533	SLOTHFULMEDIA	SLOTHFULMEDIA has created a service on victim machines named "TaskFrame" to establish persistence.[121]
S1037	STARWHALE	STARWHALE has the ability to create the following Windows service to establish persistence on an infected host: `sc create Windowscarpstss binpath= "cmd.exe /c cscript.exe c:\\windows\\system32\\w7_1.wsf humpback_whale" start= "auto" obj= "LocalSystem"`.[122]
S0142	StreamEx	StreamEx establishes persistence by installing a new service pointing to its DLL and setting the service to auto-start.[123]

ID	Name	Description
S0491	StrongPity	StrongPity has created new services and modified existing services for persistence.[124]
S0603	Stuxnet	Stuxnet uses a driver registered as a boot start service as the main load-point.[125]
S1049	SUGARUSH	SUGARUSH has created a service named `Service1` for persistence.[126]
S0663	SysUpdate	SysUpdate can create a service to establish persistence.[95]
S0164	TDTESS	If running as administrator, TDTESS installs itself as a new service named bmwappushservice to establish persistence.[127]
G0139	TeamTNT	TeamTNT has used malware that adds cryptocurrency miners as a service.[128]
S0560	TEARDROP	TEARDROP ran as a Windows service from the `c:\windows\syswow64` folder.[129][130]
G0027	Threat Group-3390	Threat Group-3390's malware can create a new service, sometimes naming it after the config information, to gain persistence.[131][132]
S0665	ThreatNeedle	ThreatNeedle can run in memory and register its payload as a Windows service.[133]
S0004	TinyZBot	TinyZBot can install as a Windows service for persistence.[134]
S0266	TrickBot	TrickBot establishes persistence by creating an autostart service that allows it to run whenever the machine boots.[135]
G0081	Tropic Trooper	Tropic Trooper has installed a service pointing to a malicious DLL dropped to disk.[136]
S0263	TYPEFRAME	TYPEFRAME variants can add malicious DLL modules as new services.TYPEFRAME can also delete services from the victim's machine.[137]
S0022	Uroburos	Uroburos has registered a service, typically named `WerFaultSvc`, to decrypt and find a kernel driver and kernel driver loader to maintain persistence.[138]
S0386	Ursnif	Ursnif has registered itself as a system service in the Registry for automatic execution at system startup.[139]
S0180	Volgmer	Volgmer installs a copy of itself in a randomly selected service, then overwrites the ServiceDLL entry in the service's Registry entry. Some Volgmer variants also install .dll files as services with names generated by a list of hard-coded strings.[140][141][142]
S0366	WannaCry	WannaCry creates the service "mssecsvc2.0" with the display name "Microsoft Security Center (2.0) Service."[143][144]
S0612	WastedLocker	WastedLocker created and established a service that runs until the encryption process is complete.[145]
S0206	Wiarp	Wiarp creates a backdoor through which remote attackers can create a service.[146]
S0176	Wingbird	Wingbird uses services.exe to register a new autostart service named "Audit Service" using a copy of the local lsass.exe file.[147][148]
S0141	Winnti for Windows	Winnti for Windows sets its DLL file as a new service in the Registry to establish persistence.[149]
G0102	Wizard Spider	Wizard Spider has installed TrickBot as a service named ControlServiceA in order to establish persistence.[150][151]
S0230	ZeroT	ZeroT can add a new service to ensure PlugX persists on the system when delivered as another payload onto the system.[102]

ID	Name	Description
S0086	ZLib	ZLib creates Registry keys to allow itself to run as various services.[152]
S0350	zwShell	zwShell has established persistence by adding itself as a new service.[153]
S0412	ZxShell	ZxShell can create a new service using the service parser function ProcessScCommand.[154]

Mitigations

ID	Mitigation	Description
M1047	Audit	Use auditing tools capable of detecting privilege and service abuse opportunities on systems within an enterprise and correct them.
M1040	Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent an application from writing a signed vulnerable driver to the system.[155] On Windows 10 and 11, enable Microsoft Vulnerable Driver Blocklist to assist in hardening against third party-developed service drivers.[156]
M1045	Code Signing	Enforce registration and execution of only legitimately signed service drivers where possible.
M1028	Operating System Configuration	Ensure that Driver Signature Enforcement is enabled to restrict unsigned drivers from being installed.
M1018	User Account Management	Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service configurations.

Detection

ID	Data Source	Data Component	Detects
DS0017	Command	Command Execution	Monitor processes and command-line arguments for actions that could create or modify services. Command-line invocation of tools capable of adding or modifying services may be unusual, depending on how systems are typically used in a particular environment. Services may also be modified through Windows system management tools such as Windows Management Instrumentation and PowerShell, so additional logging may need to be configured to gather the appropriate data. Also collect service utility execution and service binary path arguments used for analysis. Service binary paths may even be changed to execute commands or scripts.
DS0027	Driver	Driver Load	Monitor for new service driver installations and loads (ex: Sysmon Event ID 6) that are not part of known software update/patch cycles. Note: Sysmon Event ID 6 (driver load) provides information on whether the loaded driver was signed with a valid signature (via the `Signature` and `SignatureStatus` fields). As such, one way to help reduce the volume of alerts and false positives associated with this event is to filter and exclude any driver load events signed by common and legitimate publishers like Microsoft.
DS0022	File	File Metadata	Adversaries may modify the binary file for an existing service to achieve Persistence while potentially Defense Evasion. If a newly created or modified runs as a service, it may indicate APT activity. However, services are frequently installed by legitimate software. A well-tuned baseline is essential to differentiating between benign and malicious service modifications. Look for events where a file was created and then later run as a service. In these cases, a new service has been created or the binary has been modified. Many programs, such as msiexec.exe, do these behaviors legitimately and can be used to help validate legitimate service creations/modifications.
DS0029	Network Traffic	Network Traffic Flow	Monitor for several ways that code can execute on a remote host. One of the most common methods is via the Windows Service Control Manager (SCM), which allows authorized users to remotely create and modify services. Several tools, such as PsExec, use this functionality. When a client remotely communicates with the Service Control Manager, there are two observable behaviors. First, the client connects to the RPC Endpoint Mapper over 135/tcp. This handles authentication, and tells the client what port the endpoint—in this case the SCM—is listening on. Then, the client connects directly to the listening port on services.exe. If the request is to start an existing service with a known command line, the the SCM process will run the corresponding command. This compound behavior can be detected by looking for services.exe receiving a network connection and immediately spawning a child process.
DS0009	Process	OS API Execution	Monitor for API calls that may create or modify Windows services (ex: `CreateServiceW()`) to repeatedly execute malicious payloads as part of persistence.

ID	Data Source	Data Component	Detects
		Process Creation	Suspicious program execution through services may show up as outlier processes that have not been seen before when compared against historical data. Look for abnormal process call trees from known services and for execution of other commands that could relate to Discovery or other adversary techniques. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement. Windows runs the Service Control Manager (SCM) within the process services.exe. Windows launches services as independent processes or DLL loads within a svchost.exe group. To be a legitimate service, a process (or DLL) must have the appropriate service entry point SvcMain. If an application does not have the entry point, then it will timeout (default is 30 seconds) and the process will be killed. To survive the timeout, adversaries and red teams can create services that direct to cmd.exe with the flag /c, followed by the desired command. The /c flag causes the command shell to run a command and immediately exit. As a result, the desired program will remain running and it will report an error starting the service. This analytic will catch that command prompt instance that is used to launch the actual malicious executable. Additionally, the children and descendants of services.exe will run as a SYSTEM user by default. Note: Create a baseline of services seen over the last 30 days and a list of services seen today. Remove services in the baseline from services seen today, leaving a list of new services. Returns all processes named cmd.exe that have services.exe as a parent process. Because this should never happen, the /c flag is redundant in the search. Analytic 2 - Services launching CMD `(sourcetype=WinEventLog:Microsoft-Windows-Sysmon/Operational EventCode="1") OR (sourcetype=WinEventLog:Security EventCode="4688") Image="cmd.exe" and ParentImage="services.exe"`
DS0019	Service	Service Creation	Creation of new services may generate an alterable event (ex: Event ID 4697 and/or 7045 [157][158]), especially those associated with unknown/abnormal drivers. New, benign services may be created during installation of new software. Analytic 1 - Creation of new services with unusual directory paths such as temporal files in APPDATA `(sourcetype=WinEventLog:Security EventCode="4697") OR (sourcetype=WinEventLog:System EventCode="7045") | where ServiceFilePath LIKE "%APPDATA%" OR ServiceImage LIKE "%PUBLIC%"`
		Service Modification	Monitor for changes made to Windows services to repeatedly execute malicious payloads as part of persistence.
DS0024	Windows Registry	Windows Registry Key Creation	Monitor for new constructed windows registry keys that may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. Analytic 1 - Creation of the HKLM\System\CurrentControlSet\Services Registry key `sourcetype=WinEventLog:Microsoft-Windows-Sysmon/Operational EventCode="12" TargetObject="HKLM\System\CurrentControlSet\Services*"`

ID	Data Source	Data Component	Detects
		Windows Registry Key Modification	Look for changes to service Registry entries that do not correlate with known software, patch cycles, etc. Service information is stored in the Registry at `HKLM\SYSTEM\CurrentControlSet\Services`. Changes to the binary path and the service startup type changed from manual or disabled to automatic, if it does not typically do so, may be suspicious. Tools such as Sysinternals Autoruns may also be used to detect system service changes that could be attempts at persistence.[159] Analytic 1 - Modification of the HKLM\System\CurrentControlSet\Services Registry key `(sourcetype=WinEventLog:Microsoft-Windows-Sysmon/Operational EventCode IN (13, 14) EventType= "SetValue" TargetObject="HKLM\System\CurrentControlSet\Services*" | where RegistryKeyPath LIKE "%ImagePath%" OR RegistryKeyPath LIKE "%Type%" OR RegistryKeyPath LIKE "%DisplayName%" OR RegistryKeyPath LIKE "%Objectname%"`