






Product 


Solutions 

Resources 

Open Source 


Enterprise 

Pricing





Sign in


Sign up

 redcanaryco / AtomicTestHarnesses


Public


 Notifications


 Fork 46


 Star 251


<> Code


 Issues


 Pull requests 2


 Actions


 Projects


 Wiki


 Security


 Insights


 Files


 7e1e4da





 Go to file


>  .github


>  TestHarnesses


>  T1055.002_PortableExecutable...


>  T1055_ProcessInjection


>  T1059.001_PowerShell


>  T1127.001_MSBuild


>  T1134.001_TokenImpersonation


>  T1134.002_CreateProcessWithT...


>  T1134.004_ParentPIDSpoofting


>  T1218.001_CompiledHTMLFile


>  T1218.005_Mshta


>  T1218.007_Msiexec


>  T1218_SignedBinaryProxyExecut...


 InvokeRemoteFXvGPUDisable...


 InvokeRemoteFXvGPUDisable...


>  T1543.003_WindowsService


>  T1574.012_COR_PROFILER

>  Tests


 AtomicTestHarnesses.psd1


 AtomicTestHarnesses.psm1


 LICENSE

 Readme.md

AtomicTestHarnesses / TestHarnesses / T1218_SignedBinaryProxyExecution / InvokeRemoteFXvGPUDisablementCommand.ps1



 mgraeber-rc Adding Invoke-ATHRemoteFXvGPUDisablementCommand 2f65db8 · 4 years ago


 History


Code


Blame

292 lines (210 loc) · 14.4 KB

Raw







1function Invoke-ATHRemoteFXvGPUDisablementCommand {

2<#

3.SYNOPSIS

4

5Executes PowerShell code using RemoteFXvGPUDisablement.exe as a proxy executable.

6

7Technique ID: T1218 (Signed Binary Proxy Execution)

8

9.DESRIPTION

10

11Invoke-ATHRemoteFXvGPUDisablementCommand executes supplied PowerShell code using Remote

12

13One of the PowerShell functions called by RemoteFXvGPUDisablement.exe is Get-VMRemoteFX

14

15Invoke-ATHRemoteFXvGPUDisablementCommand is used to demonstrate how a PowerShell host e

16

17Note: This technique will not work under the following conditions:

18

191. RemoteFXvGPUDisablement.exe is not present.

202. PowerShell Constrained Language Mode is enforced. Because the temporary module writt

21

22.PARAMETER RemoteFXvGPUDisablementFilePath

23

24Specifies an alternate directory to execute RemoteFXvGPUDisablement.exe from. if -Remot

25

26.PARAMETER ScriptBlock

27

28Specifies optional PowerShell code to execute. Note that supplied PowerShell code will

29

30.PARAMETER ModuleName

31

32Specifies a temporary module name to use. If -ModuleName is not supplied, a 16-characte

33

34.PARAMETER ModulePath

35

36Specifies an alternate, non-default PowerShell module path for RemoteFXvGPUDisablement.

37

38.PARAMETER TestGuid

39

40Optionally, specify a test GUID value to use to override the generated test GUID behavi

41

42.OUTPUTS

43

44PSObject

45

46Outputs an object consisting of relevant execution details. The following object proper

47

48* TechniqueID - Specifies the relevant MITRE ATT&CK Technique ID.

49* TestSuccess - Will be set to True if it was determined that the PowerShell code succe

50* TestGuid - Specifies the test GUID that was used for the test.

51* ModulePath - Specifies the path to the temporary module created.

52* ModuleContents - Specifies the contents of the custom implementation of the Get-VMRem

53* ModuleFileHash - Specifies the SHA256 file hash of the custom script module file.

54* RunnerFilePath - Specifies the full path of RemoteFXvGPUDisablement.exe.

55* RunnerProcessId - Specifies the process ID of RemoteFXvGPUDisablement.exe.

56* RunnerCommandline - Specifies the command line of RemoteFXvGPUDisablement.exe.

Page 1 of 5

```
56     * RunnerCommandLine - Specifies the command-line of RemoteFXvGPUDisablement.exe.
57     * RunnerChildProcessId - Specifies the process ID of the process that was executed as t
58     * RunnerChildProcessCommandLine - Specifies the command-line of process that was execut
59
60     .EXAMPLE
61
62     Invoke-ATHRemoteFXvGPUDisablementCommand
63
64     .EXAMPLE
65
66     Invoke-ATHRemoteFXvGPUDisablementCommand -ScriptBlock { Get-Date | Out-File -FilePath '
67
68     .EXAMPLE
69
70     Invoke-ATHRemoteFXvGPUDisablementCommand -ModuleName Foo
71
72     .EXAMPLE
73
74     Invoke-ATHRemoteFXvGPUDisablementCommand -ModulePath $PWD
75
76     Executes PowerShell code from a user-supplied module path, in this case, the current di
77
78     .EXAMPLE
79
80     Copy-Item -Path "$Env:windir\System32\RemoteFXvGPUDisablement.exe" -Destination 'notepa
81     Invoke-ATHRemoteFXvGPUDisablementCommand -RemoteFXvGPUDisablementFilePath 'notepad.exe'
82
83     Executes RemoteFXvGPUDisablement.exe from a relocated and renamed executable, notepad.e
84
85     .LINK
86
87     https://support.microsoft.com/en-us/help/4558998/windows-10-update-kb4558998
88     https://support.microsoft.com/en-us/help/4570006/update-to-disable-and-remove-the-remot
89     https://twitter.com/pronichkin/status/1285241439052427265
90     #>
91
92     [CmdletBinding()]
93     param (
94         [String]
95         [ValidateNotNullOrEmpty()]
96         $RemoteFXvGPUDisablementFilePath = "$Env:windir\System32\RemoteFXvGPUDisablemen
97
98         [ScriptBlock]
99         $ScriptBlock,
100
101         [String]
102         [ValidateNotNullOrEmpty()]
103         $ModuleName = ((1..16 | ForEach-Object { [Char] (Get-Random -Minimum 0x41 -Maxi
104
105         [String]
106         [ValidateScript({ Test-Path -Path $_ -PathType Container })]
107         $ModulePath,
108
109         [Guid]
110         $TestGuid = (New-Guid)
111     )
112
113     $ModuleExecuted = $null
114     $FullModulePath = $null
115     $ExecutedRemoteFXvGPUDisablementCommandLine = $null
116     $ExecutedRemoteFXvGPUDisablementPID = $null
117     $SpawnedProcCommandLine = $null
```



```
219         if ($ModulePath) {
220             # Prepend the supplied module path to %PSModulePath%
221             $CustomPSModulePath = "PSModulePath=$($FullModulePath);$($Env:PSModulePath)"
222
223             # Gather up all existing environment variables except %PSModulePath%.
224             [String[]] $AllEnvVarsExceptPSModulePath = Get-ChildItem Env:\* -Exclude 'PSMod
225
226             [String[]] $AllEnvVars = $AllEnvVarsExceptPSModulePath + $CustomPSModulePath
227
228             $ProcessStartupInstance.EnvironmentVariables = $AllEnvVars
229         }
230
231         $ProcStartResult = Invoke-CimMethod -ClassName Win32_Process -MethodName Create -Ar
232
233         if ($ProcStartResult.ReturnValue -eq 0) {
234             # Retrieve the actual command-line of the spawned PowerShell process
235             $ExecutedRemoteFXvGPUDisablementProcInfo = Get-CimInstance -ClassName Win32_Pro
236             $ExecutedRemoteFXvGPUDisablementCommandLine = $ExecutedRemoteFXvGPUDisablementP
237             $ExecutedRemoteFXvGPUDisablementPID = $ProcStartResult.ProcessId
238             $RemoteFXvGPUDisablementFullPath = $ExecutedRemoteFXvGPUDisablementProcInfo.Exe
239         } else {
240             Write-Error "RemoteFXvGPUDisablementFullPath.exe child process was not spawned.
241         }
242
243         if (-not $ScriptBlock) {
244             # Wait for the test powershell.exe execution to run
245             $ChildProcSpawnedEvent = Wait-Event -SourceIdentifier 'ChildProcSpawned' -Timeo
246             $ChildProcInfo = $null
247
248             if ($ChildProcSpawnedEvent) {
249                 $ModuleExecuted = $True
250
251                 $ChildProcInfo = $ChildProcSpawnedEvent.MessageData
252                 $SpawnedProcCommandLine = $ChildProcInfo.ProcessCommandLine
253                 $SpawnedProcProcessId = $ChildProcInfo.ProcessId
254
255                 $ChildProcSpawnedEvent | Remove-Event
256             } else {
257                 Write-Error "powershell.exe child process was not spawned."
258             }
259
260             # Cleanup
261             Unregister-Event -SourceIdentifier 'ProcessSpawned'
262         }
263
264         [PSCustomObject] @{
265             TechniqueID      = 'T1218'
266             TestSuccess      = $ModuleExecuted
267             TestGuid         = $TestGuid
268             ModulePath       = $ModuleScriptPath
269             ModuleContents   = $FunctionToExecute
270             ModuleFileHash   = $ScriptModuleFileHash
271             RunnerFilePath   = $RemoteFXvGPUDisablementFullPath
272             RunnerProcessId  = $ExecutedRemoteFXvGPUDisablementPID
273             RunnerCommandLine = $ExecutedRemoteFXvGPUDisablementCommandLine
274             RunnerChildProcessId      = $SpawnedProcProcessId
275             RunnerChildProcessCommandLine = $SpawnedProcCommandLine
276         }
277
278         # Sleep a few seconds to give it some time to execute prior to deleting the tempora
279         if ($ScriptBlock) {
```

```
280         Start-Sleep -Seconds 2
281     }
282
283     # Delete the module that was just created
284     Write-Verbose "Deleting the script module: $ModuleScriptPath"
285     Remove-Item -Path $ModuleScriptPath -Force -ErrorAction SilentlyContinue
286
287     Write-Verbose "Deleting the module path: $NewModulePath"
288     Remove-Item -Path $NewModulePath -Force -ErrorAction SilentlyContinue
289
290     Remove-Module -ModuleInfo $ModuleInfo -ErrorAction SilentlyContinue
291     Remove-Item Function:\Get-VMRemoteFXPhysicalVideoAdapter -ErrorAction SilentlyContinue
292 }
```